

THESE DE DOCTORAT
DE L'UNIVERSITE PARIS I – PANTHEON - SORBONNE

Spécialité : Informatique

Kadan ALJOUMAA

Pour l'obtention du titre de :

DOCTEUR DE L'UNIVERSITE PARIS I – PANTHEON - SORBONNE

Modélisation intentionnelle et annotation
sémantique pour la réutilisation de services métiers

PASIS: Publishing And Searching of Intentional Services

Soutenue publiquement le **20 octobre 2011** devant le jury composé de

M. Camille SALINESI	Président
Mme Colette ROLLAND	Directeur de thèse
M. Saïd ASSAR	Co-directeur de thèse
Mme Carine SOUVEYET	Co-directeur de thèse
M. Samir TATA	Rapporteur
M. Naoufel KRAIEM	Rapporteur

À mon épouse bien aimée Katia,

à notre fils, le sage, Itadi,

et à notre fille, la joyeuse, Farah

REMERCIEMENTS

Certaines rencontres donnent l'énergie, psychologique ou financière, pour continuer un projet. A ce titre, je tiens à remercier les personnes qui m'ont soutenu pendant ce projet.

Tout d'abord, j'adresse mes sincèrement remerciement à Colette Rolland, Professeur à l'Université de Paris 1 Panthéon - Sorbonne pour la confiance qu'elle m'a témoignée en m'accueillant dans son équipe et en acceptant la direction scientifique de mes travaux. Ses idées, son savoir-faire, ses précieux conseils et sa rigueur scientifique ont fortement contribué à ce travail de recherche. J'ai beaucoup appris à ses côtés et je lui en suis reconnaissant.

Je remercie vivement Saïd Assar, Maître de Conférences à Télécom Ecole de Management à Evry pour son aide constante, ses encouragements, ses conseils et son efficacité pendant cette thèse. Sa contribution dans l'encadrement de ce travail de recherche m'a été d'une aide très précieuse. Sa disponibilité, ses encouragements et son investissement, notamment dans les moments de doute, m'ont permis de continuer à mener l'ensemble de ce travail à bien.

Je remercie vivement Carine Souveyet, Professeur à l'Université de Paris 1 Panthéon - Sorbonne pour son soutien et son aide constante. Ses encouragements, ses conseils et son efficacité m'ont été très utiles pendant ces années de travail en commun.

Je remercie Camille Salinesi, Professeur à l'Université de Paris 1 Panthéon - Sorbonne pour avoir cordialement accepté de présider le jury de cette thèse.

Je souhaite remercier chaleureusement les rapporteurs de ma thèse, Monsieur Samir Tata, Professeur à Télécom Sud Paris à Evry, et Monsieur Naoufel Kraiem, Professeur à l'Institut Supérieur d'Informatique en Tunisie, pour le travail que leur a demandé mon rapport de thèse, pour leur jugement sur ces travaux, et les perspectives de recherche qu'ils m'ont proposées.

Je remercie également Wael Khansa, Professeur et Directeur à l'ISSAT à Damas, et Iyad Zoukar, Directeur de la coopération nationale à l'ISSAT, pour leurs soutiens pendant ces trois années.

Je remercie tous les membres de l'équipe du Centre de Recherche en Informatique, et tout particulièrement Daniel, Manuele, Rebecca, Irina, Charlotte, Assia, Eléna, Islem, Kahina, Oumaima, Salma, Sana, Adrian, Bruno, Hicham, Ramzi et Raul, pour leur collaboration, leur gentillesse et leur soutien.

Enfin, je tiens aussi à remercier ma famille, mes proches et mes amis pour leur soutien.

RESUME

Ce travail est une continuation des travaux antérieurs sur l'architecture iSOA et le modèle de service intentionnel MiS. Il s'agit de rendre accessible les services proposés par MiS en complétant la plateforme iSOA avec des mécanismes pour la publication et la recherche de services intentionnels dans un annuaire étendu. Ces mécanismes feront appel aux techniques des services web sémantiques et à la modélisation de la connaissance sous forme d'ontologies.

Le propos de ce travail de recherche porte sur les lacunes identifiées sur le processus de publication d'un service, l'approche de recherche adéquate pour sélectionner les services intentionnels les plus adaptés aux besoins de l'utilisateur et la structure du descripteur d'un service intentionnel à publier dans l'annuaire.

Nous proposons l'approche *PASIS* (**P**ublishing **A**nd **S**earching **i**ntentional **S**ervices). Dans cette approche, l'input d'un fournisseur lors de la publication d'un service s'exprime en termes métier sous forme de buts. Nous proposons un processus de guidage de publication, pour d'une part, aider le concepteur dans cette tâche, et d'autre part garantir la complétude de la description d'un service publié. Ce processus utilise les ontologies pour capturer la sémantique du descripteur de service intentionnel. Ces ontologies sont : *iSOnto* (ontologie technique de services intentionnels), *vOnto* (ontologie générique des verbes) et *pOnto* (ontologie de domaine des produits). Enfin, notre approche prend en compte au niveau du descripteur et du processus de publication, les agrégations de services intentionnels telles que définies dans le modèle de MiS.

Dans *PASIS*, l'utilisateur exprime ses besoins sous forme de but. Grâce à un modèle du processus de recherche et d'interrogation, nous proposons le guidage pour d'une part, aider l'utilisateur à tirer profit de toutes les possibilités qu'offre le modèle de requête, et d'autre part pour l'accompagner dans sa recherche et l'aider à trouver ce qu'il souhaite. L'appariement se fait entre les concepts associés à la requête et ceux associés aux descripteurs de services en utilisant une fonction de calcul de distance sémantique qui exploite les liens sémantiques (généralisation, spécialisation). En exploitant les concepts de services intentionnels agrégés, l'utilisateur peut parcourir l'arborescence de services composites et à variation et explorer ainsi l'espace des services intentionnels disponibles. Enfin, l'usage d'ontologies est la base d'un processus de reformulation intelligente de la requête de l'utilisateur. L'utilisation de l'ontologie générique des verbes et des liens sémantiques entre les verbes et les noms propre, nous pouvons raisonner sur la signification d'un but et d'introduire ainsi intelligence et guidage dans les processus de publication et de recherche de service.

Ce travail est consacré également, à la spécification et à l'implémentation du descripteur intentionnel de service où l'annotation est utilisée comme une technique d'extension sémantique vers les trois ontologies : *iSOnto*, *vOnto* et *pOnto*. Dans ce travail, nous proposons d'étendre le langage SAWSDL pour décrire les services intentionnels. Cette extension vient combler un manque au niveau de la spécification de la nature de l'annotation sémantique dans SAWSDL. Pour cette raison, nous allons utiliser l'ontologie *iSOnto* pour expliciter non seulement les concepts d'un domaine mais également la nature de ces concepts.

Enfin, l'approche proposée est illustrée au moyen d'une étude de cas d'application. A la lumière de ce cas d'étude, nous avons montré que les hypothèses de recherche sont valides.

ABSTRACT

This work is a continuation of previous work on architecture and iSOA and the intentional service model MiS. This is to make the services provided by MiS available by completing the platform iSOA. In other words, publishing and searching intentional services in an extended registry. The mechanisms used involve semantic web services techniques and knowledge modelling in form of ontologies.

The purpose of this work focuses on the gaps identified in the process of publishing a service, the discovery approach to select the most appropriate intentional services to satisfy user's needs and the structure of the intentional service descriptor to be published in the registry.

We offer the approach *PASiS* (**P**ublishing **A**nd **S**earching intentional **S**ervices). In this approach, the provider input in the publication of a service is expressed in business terms as goals. We propose a guided publication process to assist the designer in this task, and also to guarantee the completeness of the published service description. This process uses ontologies to capture the semantics of the intentional service descriptor. These ontologies are *iSOnto* (**i**ntentional **S**ervices **O**ntology), *vOnto* (**v**erbs **O**ntology) and *pOnto* (**p**roducts **O**ntology). Finally, our approach takes into account, in the descriptor and in the publishing process, the aggregation of intentional services as defined in the model MiS.

In *PASiS*, the user expresses his needs as a goal. Through a discovery process model, we offer guidance to help the user take advantage of all opportunities in the query template, and also to guide him in his discovery and help him find what he wants. The matching is done between the concepts associated with the query and those associated with the service descriptors, using a semantic distance calculation function that exploits the semantic relations (generalization, specialization) in domain ontology. By exploiting the concepts of intentional aggregated services, the user can browse the tree of composite/variant services to explore the space of intentional services available. Finally, the smart reformulation process of user's request is based on the use of ontologies. By using of generic ontology of verbs and semantic links between verbs and nouns, we can reason on the meaning of a goal and introduce intelligence and guidance in the publishing and searching processes.

This work is devoted also to the specification and implementation of intentional descriptor of service where the annotation is used as a technique for semantic extension to the three ontologies: *iSOnto*, *vOnto* and *pOnto*. In this work, we propose to extend the SAWSDL language to describe intentional services. This extension fills a gap in the specification of the nature of semantic annotation in SAWSDL. For this reason, we will use the ontology *iSOnto* to explain not only the concepts of a domain but also the nature of these concepts.

Finally, the proposed approach is illustrated through a case study application. In light of this case study, we showed that the research hypotheses are valid.

Table des matières

CHAPITRE 1 : INTRODUCTION	17
1. Contexte.....	17
1.1. Les services web et l'architecture SOA.....	17
1.2. Les services web sémantiques.....	18
1.3. Le modèle MiS et l'approche iSOA.....	20
1.4. Motivation de recherche	21
2. Problématique.....	22
2.1. L'expression au niveau métier des besoins du fournisseur	22
2.2. L'expression au niveau métier des besoins du client.....	24
2.3. Le besoin d'une sémantique étendue	25
3. Aperçu de la solution.....	26
3.1. Hypothèses.....	27
3.2. Contributions	27
4. Plan du mémoire.....	28
CHAPITRE 2 : ETAT DE L'ART	29
1. Introduction	29
1. Cadre de référence pour les services	29
1.1. Méta-modèle utilisé	30
1.2. Description du cadre de référence	31
1.3. La vue sujet.....	31
1.3.1. L'entité	32
1.3.2. Granularité.....	32
1.3.3. Visibilité	33
1.4. La vue usage	34
1.4.1. Finalités	34
1.4.2. Fonctionnalités	34
1.4.3. Phase.....	35
1.5. La vue modèles	36
1.5.1. Modèle de produit.....	36
1.5.1.1. Modèle input utilisateur :.....	36
1.5.1.2. Modèle de requête	37
1.5.1.3. Modèle de descripteur	37
1.5.1.4. Modèle input fournisseur.....	38
1.5.1.5. Nature d'ontologie.....	38

Table de matières

1.5.2.	Modèle de processus.....	39
1.5.2.1.	Processus de publication.....	39
1.5.2.2.	Processus de formulation de requête.....	40
1.5.2.3.	Processus d'appariement	40
1.6.	<i>La vue implémentation</i>	41
1.6.1.	Technique de mesure	41
1.6.2.	Usage d'ontologie.....	42
1.6.3.	L'architecture logicielle.....	42
1.7.	<i>Résumé du cadre de référence</i>	43
2.	Positionnement des approches au moyen du cadre de référence.....	45
2.1.	<i>L'approche GODO [Gomez et al., 2006]</i>	45
2.2.	<i>L'approche SAWSDL-MX [Klusch et al., 2009]</i>	47
2.3.	<i>L'approche SeCSE [Zachos et al., 2007]</i>	49
2.4.	<i>L'approche de [Da Silva et al., 2009]</i>	51
2.5.	<i>L'approche YASA [Chabeb et al., 2009]</i>	54
2.6.	<i>L'approche SATIS [Mirbel et Crescenzo, 2009]</i>	55
2.7.	<i>L'approche de [Driss et al., 2010]</i>	58
3.	Récapitulatif de l'évaluation	60
4.	Conclusion.....	63
CHAPITRE 3 :	L'APPROCHE PASiS	65
1.	Introduction	65
2.	L'approche PASiS	66
2.1.	<i>Rappel de la problématique</i>	66
2.2.	<i>Vue globale de l'approche</i>	66
3.	Caractéristiques de l'approche	69
3.1.	<i>Expression orientée métier</i>	69
3.2.	<i>Le guidage dans les processus</i>	71
3.2.1.	Processus de publication.....	72
3.2.2.	Processus de recherche	72
3.3.	<i>Utilisation d'ontologies</i>	72
3.4.	<i>Usage du degré de similarité</i>	73
4.	Ontologies dans PASiS	74
4.1.	<i>L'ontologie de services intentionnels (iSOnto)</i>	74
4.2.	<i>L'ontologie des verbes vOnto</i>	76
4.2.1.	Types de verbes	76
4.2.2.	Catégories de verbes.....	77
4.2.3.	Signature de verbes.....	77
4.3.	<i>L'ontologie de domaine pOnto</i>	78
4.4.	<i>Relation entre vOnto et pOnto</i>	79
5.	Extension sémantique du descripteur de service.....	80

5.1.	<i>L'usage de l'annotation</i>	81
5.2.	<i>L'approche YASA</i>	81
5.3.	<i>Le descripteur intentionnel</i>	82
5.4.	<i>Exemple du descripteur intentionnel</i>	85
5.4.1.	Représentation des services intentionnels dans iSonto	85
5.4.2.	Descripteurs des services intentionnels en SAWSDL	87
6.	Modèle de Carte	88
6.1.	<i>Méta-modèle de processus</i>	89
6.2.	<i>Principe de guidage</i>	90
6.2.1.	Notion de directive	91
6.2.2.	Signature.....	91
6.2.2.1.	Situation d'une signature	92
6.2.2.2.	Intention d'une signature	92
6.2.3.	Corps	92
6.3.	<i>Typologie des directives</i>	93
6.3.1.	Directive stratégique.....	93
6.3.2.	Directive tactique.....	93
6.3.2.1.	Directive choix	94
6.3.2.2.	Directive plan	95
6.3.2.3.	Directive exécutable	96
6.3.2.4.	Hiérarchie de directives tactiques	96
6.3.3.	Directive informelle.....	97
6.4.	<i>Directives associées à la Carte</i>	97
6.4.1.	Directive de réalisation d'intention	98
6.4.2.	Directive de sélection de stratégie	98
6.4.3.	Directive de sélection d'intention	99
7.	Conclusion	100
CHAPITRE 4 : PUBLICATION DE SERVICES INTENTIONNELS		101
1.	Introduction	101
2.	Etape préalable d'identification des services	101
2.1.	<i>Utilisation de la Carte</i>	102
2.2.	<i>La découverte des services intentionnels</i>	103
2.2.1.	Le modèle intentionnel de services.....	104
2.2.2.	L'identification des services intentionnels à partir de la carte.....	105
3.	Processus de publication	106
3.1.	<i>Sommaire des directives du Map-Publication</i>	107
3.2.	<i>Exploration du MAP-Publication</i>	108
3.2.1.	Réaliser « Annoter le service Par caractérisation des éléments de base » (DRI ₁).....	108
3.2.2.	Progresser vers « Annoter le service » (DSS ₁)	109
3.2.2.1.	Réaliser « Annoter le service par annotation du verbe » (DRI ₂)	110
3.2.2.2.	Réaliser « Annoter le service par annotation d'objet » (DRI ₃)	111

Table de matières

3.2.2.3.	Réaliser « Annoter le service par annotation des agrégats » (DRI ₄)	111
3.2.2.4.	Réaliser « Annoter le service par annotation des paramètres » (DRI ₅)	111
3.2.2.5.	Réaliser « Annoter le service par raisonnement sur les états » (DRI ₆)	112
3.2.2.5.1.	Réaliser « raisonner sur les verbes causatifs » (DRI _{6.1})	112
3.2.2.5.1.1.	Réaliser « raisonner sur les verbes causatifs physiques » (DRI _{6.1.1})	113
3.2.2.5.1.2.	Réaliser « raisonner sur les verbes causatifs intellectuels » (DRI _{6.1.1})	115
3.2.2.5.2.	Réaliser « raisonner sur les verbes descriptifs » (DRI _{6.2})	116
3.2.2.6.	Réaliser « Annoter le service par résolution des conflits » (DRI ₇)	116
3.2.3.	Réaliser « Annoter le service Par vérification » (DRI ₈)	118
3.2.3.1.	Réaliser « vérifier un élément de l'annotation » (DRI _{8.1})	118
3.2.3.2.	Réaliser « valider les annotations » (DRI _{8.2})	119
4.	Conclusion	119
CHAPITRE 5 : RECHERCHE DE SERVICES ET FORMULATION DE REQUÊTES		121
1.	Introduction	121
2.	Le langage de requêtes	121
3.	La reformulation et le degré de similarité	122
3.1.	<i>L'algorithme d'appariement</i>	<i>123</i>
3.2.	<i>Exemple d'utilisation de l'algorithme d'appariement</i>	<i>125</i>
4.	Processus de reformulation de requêtes	127
4.1.	<i>Sommaire des directives du Map-Recherche</i>	<i>130</i>
4.2.	<i>Exploration du Map-Recherche</i>	<i>131</i>
4.2.1.	Progresser vers « Formuler une requête » (DSS ₁)	131
4.2.1.1.	Réaliser « Formuler une requête Par langage naturel » (DRI ₁)	132
4.2.1.2.	Réaliser « Formuler une requête Par Template » (DRI ₂)	133
4.2.2.	Progresser depuis « Formuler une requête » (DSI ₁)	134
4.2.2.1.	Réaliser « Proposer des services Par recherche » (DRI ₃)	136
4.2.2.2.	Réaliser « Arrêter Par abandon » (DRI ₄)	136
4.2.2.3.	Progresser vers « Formuler une requête » (DSS ₂)	136
4.2.2.3.1.	Réaliser « Formuler une requête Par structuration selon le template » (DRI ₅)	138
4.2.2.3.2.	Réaliser « Formuler une requête Par mise en adéquation du verbe et de l'objet » (DRI ₆)	138
4.2.2.3.3.	Réaliser « Formuler une requête Par glissement de verbe » (DRI ₇)	139
4.2.2.3.4.	Réaliser « Formuler une requête Par complétude de l'énoncé de but » (DRI ₈)	140
4.2.2.3.5.	Réaliser « Formuler une requête Par décomposition (sous buts) » (DRI ₉)	141
4.2.2.3.6.	Réaliser « Formuler une requête Par glissement de l'objet » (DRI ₁₀)	142
4.2.2.3.7.	Réaliser « Formuler une requête Par l'utilisation des états » (DRI ₁₁)	142
4.2.3.	Progresser depuis « Proposer des services » (DSI ₂)	143
4.2.3.1.	Réaliser « Proposer des services Par affinement » (DRI ₁₂)	145
4.2.3.2.	Progresser vers « Proposer des services » (DSS ₃)	145
4.2.3.2.1.	Réaliser « Proposer des services Par exploration des composites » (DRI ₁₃)	146
4.2.3.2.2.	Réaliser « Proposer des services Par exploration des variantes » (DRI ₁₄)	147
4.2.3.2.3.	Réaliser « Proposer des services Par exploration des composants » (DRI ₁₅)	148
4.2.3.3.	Progresser vers « Arrêter » (DSS ₄)	148

4.2.3.3.1.	Réaliser « Arrêter Par sélection » (DRI ₁₆)	149
4.2.3.3.2.	Réaliser « Arrêter Par abandon » (DRI ₁₇)	149
5.	Scénario d'utilisation	149
5.1.	<i>Principe général</i>	149
5.2.	<i>Présentation de l'exemple</i>	149
5.3.	<i>Déroulement du scénario</i>	149
6.	Conclusion	152
CHAPITRE 6 :	CAS D'APPLICATION	153
1.	Introduction	153
2.	Introduction du cas d'étude	153
2.1.	<i>Cas d'étude</i>	153
2.2.	<i>Analyse des dysfonctionnements</i>	155
3.	Publication des services intentionnels	155
3.1.	<i>La carte « e-clinique »</i>	155
3.1.1.	Intentions de la carte e-clinique	156
3.1.2.	Stratégies de la carte e-clinique	156
3.2.	<i>Identification des services intentionnels dans la carte e-clinique</i>	158
3.2.1.	Etape 1 : Définir les services atomiques	158
3.2.2.	Etape 2 : Définir les services agrégats	159
3.2.2.1.	Génération de tous les chemins de la carte	159
3.2.2.2.	Identification des services agrégats	161
3.3.1.	Caractérisation des éléments de base des services intentionnels	163
3.3.1.1.	Formulation des buts de services intentionnels	164
3.3.2.	Annotation de services intentionnels	165
3.3.2.1.	Annotation de verbe	165
3.3.2.2.	Annotation d'objet	165
3.3.2.3.	Annotation des agrégats	166
3.3.2.4.	Raisonnement sur les états	166
3.3.2.5.	Résolution des conflits	167
3.3.2.6.	Représentation des services intentionnels dans iSonto	167
3.3.3.	Vérification des annotations	169
3.4.	<i>Descripteurs des services intentionnels en SAWSDL</i>	170
4.	Recherche des services intentionnels	171
4.1.	<i>Déroulement du processus de recherche</i>	171
4.1.1.	Premier scénario	172
4.1.1.1.	Formulation de requête par template	172
4.1.1.2.	Reformulation de requête	173
4.1.1.2.1.	Formulation de requête par glissement de verbe	173
4.1.1.2.2.	Formulation de requête par glissement d'objet	173
4.1.1.2.3.	Recherche de services	174
4.1.1.3.	Proposition des services	174
4.1.1.3.1.	Proposition des services par exploration des variantes	174

Table de matières

4.1.1.3.2.	Proposition des services par exploration des composites	174
4.1.2.	Deuxième scénario	174
4.1.2.1.	Formulation de requête par template	175
4.1.2.2.	Reformulation de requête	175
4.1.2.2.1.	Formulation de requête par décomposition de but.....	175
4.1.2.2.2.	Formulation de requête par glissement de verbe	175
4.1.2.2.3.	Formulation de requête par glissement d'objet.....	175
4.1.2.2.4.	Recherche de services	176
4.1.2.3.	Proposition des services.....	176
4.1.2.3.1.	Proposition des services par exploration des composants	176
4.1.2.3.2.	Proposition des services par exploration des variantes.....	177
4.1.2.3.3.	Proposition des services par exploration des composites	177
4.1.2.3.4.	Sélection des services.....	177
5.	Discussion des hypothèses.....	178
5.1.	Rappel des hypothèses	178
5.2.	Discussion.....	178
6.	Prototype expérimental.....	179
6.1.	Architecture du prototype	179
6.2.	Prototype de PASiS.....	180
6.2.1.	Le niveau « interface ».	181
6.2.2.	Le niveau « traitement ».	181
6.2.3.	Le niveau « accès aux données ».	182
6.3.	La base de services intentionnels et les ontologies.....	182
6.3.1.	Les services intentionnels	182
6.3.2.	Les ontologies.....	183
6.4.	Illustration du prototype	184
7.	Conclusion.....	188
CHAPITRE 7 : CONCLUSION ET PERSPECTIVES.....		189
Perspectives		190
BIBLIOGRAPHIE		193
ANNEXE A		201
1.	Services intentionnels de la carte e-clinique	201
1.1.	La carte « e-clinique ».....	201
1.1.1.	Les services atomiques	201
1.1.2.	Les services agrégats	202
1.2.	Représentation des services intentionnels et descripteurs.....	203
1.2.1.	Service atomique S ₁	203
1.2.2.	Service atomique S ₂	204
1.2.3.	Service atomique S ₃	205
1.2.4.	Service atomique S ₄	206
1.2.5.	Service atomique S ₅	207

Table de matières

1.2.6.	Service atomique S_6	208
1.2.7.	Service atomique S_7	209
1.2.8.	Service atomique S_8	210
1.2.9.	Service atomique S_9	211
1.2.10.	Service atomique S_{10}	212
1.2.11.	Service atomique S_{11}	213
1.2.12.	Service atomique S_{12}	214
1.2.13.	Service atomique S_{12}	215
1.2.14.	Service atomique S_{14}	216
1.2.15.	Service atomique S_{15}	217
1.2.16.	Service atomique S_{16}	218
1.2.17.	Service atomique S_{17}	219
1.2.18.	Service atomique S_{18}	220
1.2.19.	Service atomique S_{19}	221
1.2.20.	Service atomique S_{20}	222
1.2.21.	Service agrégat S_{23}	223
1.2.22.	Service agrégat S_{24}	224
1.2.23.	Service agrégat S_{25}	225
1.2.24.	Service agrégat S_{26}	226
1.2.25.	Service agrégat S_{27}	227
1.2.26.	Service agrégat S_{28}	228
1.2.27.	Service agrégat S_{29}	229
1.2.28.	Service agrégat S_{30}	230
1.2.29.	Service agrégat S_{31}	231
1.2.30.	Service agrégat S_{32}	232
1.2.31.	Service agrégat S_{33}	233
ANNEXE B		234
1. Ontologie des services intentionnels « <i>iSOnto</i> » par Protégé.....		234
2. L'ontologie <i>iSOnto</i> en OWL.....		234

CHAPITRE 1

INTRODUCTION

Le travail réalisé dans cette thèse s'inscrit dans le domaine de l'ingénierie des systèmes d'information à base de services.

1. CONTEXTE

L'Ingénierie des Systèmes d'Information (ISI) est l'ensemble des activités qui permettent la construction de Systèmes d'Information (SI) de qualité. Cette construction est un processus qui aboutit à la production d'un logiciel et à sa mise en place dans un contexte organisationnel et humain. Cependant, la complexité croissante des systèmes et leur incessante évolution rendent leur développement plus laborieux, plus coûteux et moins fiable. La réutilisation s'inscrit comme une solution pour faire face à ces difficultés. Elle se définit comme une approche de développement de systèmes à partir de modules logiciels existants ayant été produits à l'occasion de développements antérieurs [Szyperski, 1997] [Semmak, 1998].

L'approche orientée objet et le développement de composants logiciels ont rendu cette approche possible et praticable. La réutilisation introduit alors deux perspectives dans l'ingénierie des SI : d'une part « la conception pour la réutilisation » (*design for reuse*) où le souci du concepteur est d'identifier de futurs composants réutilisables et de les rendre disponibles pour de futurs développements, et d'autre part, « la conception par la réutilisation » (*design by reuse*) qui cherche à construire des systèmes à partir de composants déjà identifiés et accessibles dans une base de composants [Assar et al., 2003] [Kung-Kiu et Zheng, 2007].

1.1. Les services web et l'architecture SOA

Les services Web et l'architecture SOA (Service Oriented Architecture) sont une extension à l'ère du web du concept de composant [Alonso et al., 2004] [Papazoglou et Van den Heuvel, 2007]. Ils sont sensés apporter une grande flexibilité dans la maintenance et l'évolution des systèmes, et garantir un haut degré d'interopérabilité entre systèmes hétérogènes [Papazoglou

et *al.*, 2008]. L'ingénierie des SI à base de service soulève des questions similaires à celle à base de composants. En effet, on trouve d'un côté les entreprises qui identifient les services et les publient afin de les rendre réutilisables par d'autres développeurs. Cette étape est réalisée par celui qui est communément appelé *fournisseur de services* (Figure 1.1). De l'autre côté, on trouve les entreprises ou les agents métiers qui souhaitent concevoir et implémenter leurs processus métiers à l'aide de services déjà existants. Ces deux facettes de l'ingénierie à base de service (fournisseur vs. utilisateur) soulèvent des besoins et des problématiques distinctes et spécifiques [Eck et Wieringa, 2003].

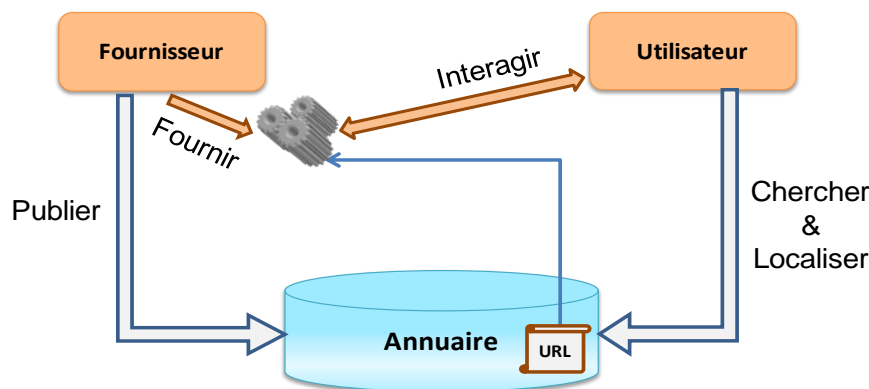


Figure 1.1. *L'architecture SOA*

Afin de rendre les services web réutilisables et interopérables, plusieurs standards ont été proposés pour unifier les moyens de publication, de localisation et d'utilisation. Il s'agit principalement du langage WSDL¹ pour la description d'un service, et les annuaires UDDI² pour le référencement. Ces outils sont sémantiquement de bas niveau et conceptuellement éloignés des préoccupations de l'utilisateur. En effet, la description des interfaces d'un service se situe au niveau technique (cf. des expressions en WSDL [WSDL, 2007]) et si cette description est compréhensible pour un ingénieur, elle l'est beaucoup moins pour un utilisateur final au niveau métier. Alors que les entreprises s'interrogent sur l'opportunité d'adopter l'architecture SOA à grande échelle [Hirschheim et *al.*, 2010] et que le concept de service est familier dans l'univers métier [Peppard, 2003], il est nécessaire d'élever le niveau d'abstraction auquel se situent la description et l'interaction avec les services web, et de chercher à combler ainsi le gap entre le service métier et le service technique. Ce point de vue est conforme avec l'analyse de plusieurs experts et praticiens de l'informatique à base de service [Zimmerman et *al.*, 2004 ; Arsanjani et Casati, 2004 ; Eckert et *al.*, 2009].

Pour enrichir la description d'un service et élever son niveau d'abstraction, plusieurs extensions ont été proposées. Nous évoquons ici, d'une part les services web sémantiques, et d'autre part, le modèle intentionnel de service qui a été le point de départ de ce travail de recherche.

1.2. Les services web sémantiques

Les services web sémantiques sont une proposition pour répondre à la problématique du faible niveau d'abstraction et de la pauvreté de description d'un service. L'association d'une

¹ WSDL : Web Service Description Language

² UDDI : Universal Description Discovery and Integration

information de nature sémantique au descripteur d'un service web a pour objectif la localisation, la composition et l'utilisation automatique de services distribués [McIlraith et al., 2001]. Cette description sémantique fait référence d'une manière ou d'une autre à une connaissance sous forme d'ontologie [Martin et Domingue, 2007a]. Une ontologie est une arborescence de concepts et de termes, structurée selon les hiérarchies de spécialisation et de généralisation. Elle doit refléter une certaine connaissance spécifique d'un domaine particulier ou générique³ lorsque les concepts et les termes sont indépendants d'un domaine.

Pour les services web sémantiques, les propositions qui ont été faites adoptent deux approches différentes pour enrichir la description d'un service :

1. La première approche définit des ontologies spécifiques pour les web services. Avec ces ontologies, le web service est décrit directement comme une instance de celles-ci. La publication d'un service revient ainsi à des opérations d'instanciation et/ou d'insertion dans l'arborescence de l'ontologie, et la recherche d'un service devient un problème de recherche dans celle-ci. Deux techniques se sont développées selon cette approche :
 - (i) Le langage de description de service OWL-S⁴ [Martin et al., 2004a et b], qui est une spécialisation du langage général de description d'ontologie OWL⁵ [Smith et al., 2004]. Cette description distingue trois types d'information essentiels pour un service : ce que fait le service (*Service Profile*), comment l'utilise-t-on (*Service Model*) et comment y accède-t-on (*Service Grounding*).
 - (ii) L'ontologie de service WSMO⁶ avec le langage de description associé WSML⁷ [Roman et al., 2005] [De Bruijn et al., 2005]. Cette technique repose sur le framework WSMF⁸ [Fensel et Bussler, 2002] qui spécifie les principaux éléments pour décrire un service web sémantique tels que la terminologie utilisée, les buts attendus du service, les fonctionnalités offertes, ainsi que les médiateurs qui lient les différents éléments afin de permettre l'interopérabilité entre les composants hétérogènes

Ces deux techniques présentent plusieurs points communs, mais restent néanmoins d'usage différents [Lara et al., 2004].

2. La seconde approche utilise des ontologies de domaine pour enrichir la description standard d'un service à l'aide d'annotations incorporées dans le descripteur du service. Cette approche donne ainsi la possibilité de faire référence à une connaissance métier relative au contexte dans lequel le service opère. La proposition SAWSDL⁹ est celle qui adopte cette approche [Farrell et Lausen, 2007]. Pour chaque élément WSDL, on peut référencer plusieurs concepts d'une ontologie de domaine dans l'attribut « *modelReference* » de SAWSDL. Cette technique a l'avantage de préserver le descripteur WSDL standard d'un service et étant proche de WSDL, SAWSDL ne nécessite pas beaucoup d'efforts pour les développeurs. La principale limitation est cependant l'absence d'information sur la nature et le type du concept qui est utilisé dans l'annotation. Ce constat est à l'origine de la proposition de YASA4WSDL qui permet d'associer une annotation métier avec un type de concept défini dans une ontologie technique de service

³ WordNet par exemple est une ontologie générique bien connue.

⁴ OWL-S : Ontology Web Language for Semantic web

⁵ OWL : Ontology Web Language

⁶ WSMO : Web Service Modeling Ontology

⁷ WSML : Web Service Modeling Language

⁸ WSMF : Web Service Modeling Framework

⁹ SAWSDL : Semantic Annotation for Web Service Description Language

[Chabeb et Tata, 2008]. L'ontologie technique fournit une connaissance sur le modèle sous-jacent au service (cf. les éléments sous-jacents à sa structure). Grâce à cette connaissance, il devient possible d'associer les annotations sémantiques à des éléments spécifiques du modèle de service. Ce qui permet de distinguer par exemple entre l'annotation sémantique que le concepteur veut associer aux paramètres d'entrée, de celle que le concepteur veut associer aux paramètres de sortie.

Malgré la référence à une certaine connaissance du domaine d'utilisation, la description enrichie d'un service sémantique reste éloignée de la vision de l'utilisateur final. En effet, les besoins de celui-ci font généralement référence à ses objectifs et à son contexte métier [Perini et *al.*, 2005]. La connaissance du service et du domaine d'activité n'est pas suffisante pour représenter la problématique métier à laquelle doit répondre un service.

1.3. Le modèle MiS et l'approche iSOA

Le Modèle intentionnel de Services MiS [Kaabi, 2007] [Kaabi et Souveyet, 2007] [Rolland et *al.*, 2008] partage avec les approches sémantiques la volonté d'enrichir la description des services web et de rencontrer ainsi les attentes des agents métiers. En effet, MiS a été proposé pour la modélisation et la description des services en termes intentionnels, en portant toute la notion d'un service au niveau des objectifs métiers de l'agent. Cette approche est dans le courant des approches d'ingénierie des exigences (telles que KAOS [Lamsweerde et Letier, 2000], TROPOS [Bresciani et *al.*, 2004], ou le MAP [Rolland et *al.*, 2008]) qui prennent explicitement en compte les objectifs métiers (fonctionnels et/ou non fonctionnels) de l'utilisateur. Elle adopte un point de vue partagée par plusieurs chercheurs selon lequel l'identification des services (ou des composants logiciels) à partir des modèles de processus métier exploite des techniques issues de l'ingénierie des exigences [Levi et Arsanjani, 2002] [Lichtenstein et *al.*, 2005] [Wang et *al.*, 2005] [Kim et *al.*, 2010].

L'approche MiS introduit ainsi un déplacement de la perspective SOA centrée 'fonction' vers une perspective équivalente qui est centrée 'intention'. Le portage de SOA vers le niveau intentionnel (qualifié d'iSOA, *intentional Service Oriented Architecture* [Rolland et *al.*, 2007]) permet la prise en compte directe des objectifs métier (Figure 1.2). Les agents métiers remplacent les agents logiciels au niveau des interactions, et la description intentionnelle du service remplace la description technique. L'agent métier qui recherche un ou plusieurs services coïncidant avec ses besoins effectue ainsi une recherche basée sur les buts métiers.

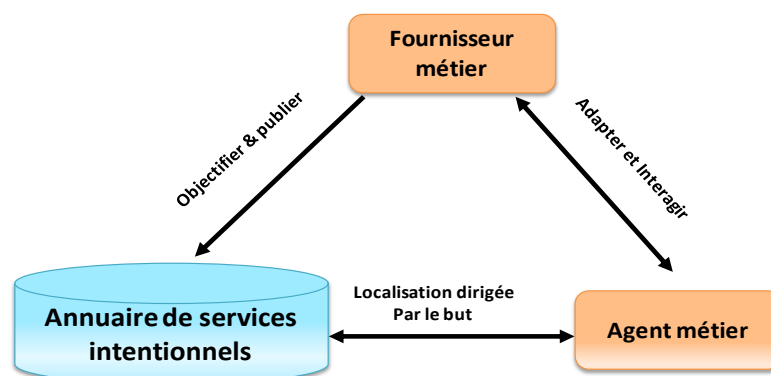


Figure 1.2. L'architecture iSOA

Dans le modèle MiS (figure 1.3), un service intentionnel correspond à un but qu'il permet de réaliser. Ce but est structuré selon une approche linguistique proposée initialement dans [Prat, 2007], et dans laquelle on distingue le verbe, le sujet et le complément. Le modèle MiS repose

sur le postulat que la richesse sémantique d'un but est capturée par l'association d'un verbe avec un sujet et des paramètres. Le sens du but dépend ainsi de la classe du verbe et des rôles que jouent les paramètres de l'expression verbale par rapport au verbe.

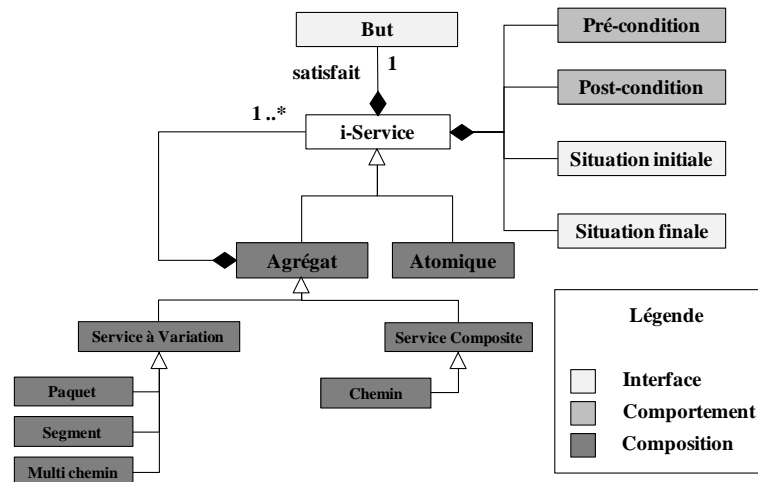


Figure 1.3. Le modèle MiS

A un service intentionnel est aussi associé un changement d'états d'un (ou plusieurs) fragments de produit(s). Ce changement d'état est décrit de deux manières : les descriptions d'une situation initiale et d'une situation finale, et l'expression d'une pré- et d'une post-condition relatives au(x) fragment(s) de produit(s) [Rolland et *al.*, 2010]. Il définit la condition formelle de réalisation du but associé au service.

Le modèle MiS apporte une grande flexibilité dans la définition d'un service intentionnel composé grâce aux différentes manières - simples ou complexes – de satisfaire un but. Un service intentionnel agrégat composite (chemin simple) ou à variation (multi-chemin, paquet, segment) correspond à un processus (ou un fragment de processus) ayant un but spécifique. Avec l'architecture iSOA, on passe ainsi d'un service web qui correspond à une opération technique, à un service intentionnel qui subsume un processus de décision sur la façon adéquate d'atteindre un but dans une situation donnée.

1.4. Motivation de recherche

Ce travail est une continuation des travaux antérieurs sur l'architecture iSOA et le modèle de service intentionnel MiS. Il prend comme point de départ le modèle MiS et la méthode d'identification de services intentionnels qui lui est associée [Kaabi, 2007]. Il s'agit de rendre opérationnelle cette proposition de recherche en complétant iSOA avec des mécanismes et des processus pour la publication et la recherche de services intentionnels dans un annuaire étendu. Ces mécanismes feront appel aux techniques des services web sémantiques et à la modélisation de la connaissance sous forme d'ontologies. La finalité de ce travail revient ainsi à tester la pertinence d'iSOA en tant qu'approche globale pour l'ingénierie à base de services orientés but.

Cet objectif de recherche soulève plusieurs questions telles que :

- (i) la structure du descripteur d'un service intentionnel : Comment faire pour préserver l'orientation but métier du service intentionnel lorsque celui-ci sera stocké dans l'annuaire ? Quelle architecture faut-il adopter pour définir un annuaire pour les services intentionnels ?

- (ii) le processus de publication d'un tel service : Comment traduire la logique métier du service intentionnel en un descripteur implémenté dans l'annuaire étendu ? Comment guider efficacement le fournisseur lors de ce processus de publication ?
- (iii) et enfin, l'approche de recherche adéquate pour sélectionner les services intentionnels les plus adaptés aux besoins de l'utilisateur : Comment formuler la requête de l'utilisateur de manière à rester le plus proche possible de la logique métier de celui-ci ? Comment aider et guider l'utilisateur dans ce processus de formulation ?

Ces questions sont discutées plus en détail dans la prochaine section.

2. PROBLEMATIQUE

Pour aborder les problématiques sous-jacentes à ce travail, on doit définir le concept de service métier au niveau duquel se situe notre réflexion. Un service métier peut être défini comme étant un groupement logique de composants techniques requis pour satisfaire une demande métier particulière [Pallos, 2001]. Il correspond à une interaction entre une organisation et un utilisateur de son système d'information [Tansey et Stroulia, 2006]. Cette interaction fait référence au contexte et aux concepts de l'activité de l'organisation, en faisant abstraction des détails techniques avec lesquels cette interaction a lieu.

Dans le modèle MiS, le but qui est associé à un service intentionnel exprime un objectif du niveau métier et l'exécution de ce service correspond à une stratégie (une manière) de réalisation de l'intention du service. L'approche iSOA se positionne ainsi au niveau des services métier et soulève de ce fait des problèmes spécifiques étant donné la difficulté déjà évoquée de la discordance entre les niveaux techniques et le niveau métier (Figure 1.4).

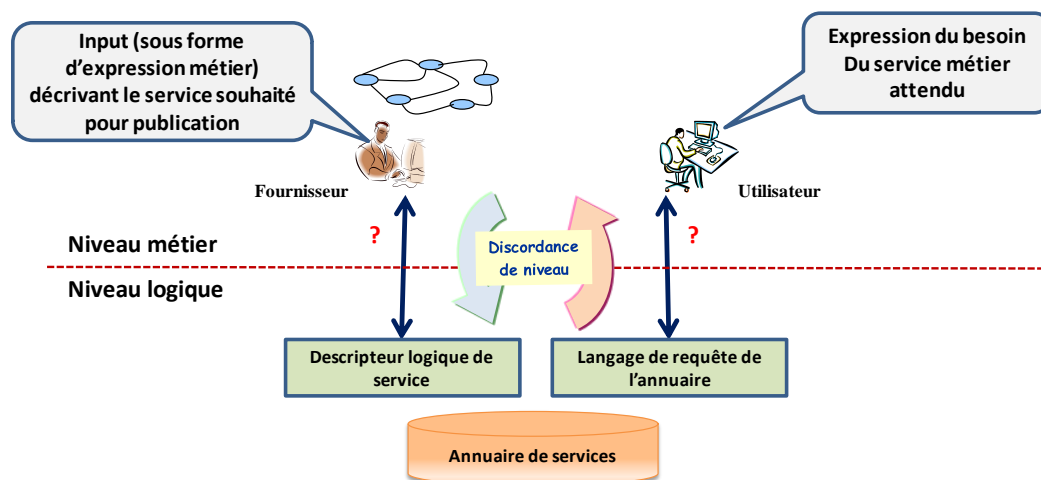


Figure 1.4. *Problème de discordance de niveau*

Par rapport aux objectifs de ce travail de recherche (cf. 1.4), le constat de discordance de niveau implique trois catégories de questions que nous détaillons dans les prochaines sous-sections.

2.1. L'expression au niveau métier des besoins du fournisseur

Du côté du fournisseur, la publication d'un service métier soulève trois questions :

- Le premier problème est relatif à la **description d'un service**. Dans les approches existantes, l'input fournisseur est directement dépendant du descripteur technique sous-

jacent. En effet, dans les approches existantes (basées sur WSDL ou SAWSDL), l'input fournisseur lors de la publication correspond à une collection de mots-clés (ayant éventuellement une sémantique) et qui sont rattachés à la structure du web service (nom, paramètres, etc.). Le processus de publication revient ainsi à une interaction directe avec le niveau technique. Alors que le fournisseur souhaite traduire de la manière la plus fidèle la logique du service qui va être publié, comment faire pour que cette traduction fasse abstraction des détails et des contraintes de l'architecture technique ?

- Le second problème est comment **faire largement connaître aux futurs clients le service** qui va être publié. Dans le cas du web documentaire, la technique du « tagging » lors de la publication d'une information permet de cibler le maximum de termes susceptibles de correspondre de près ou de loin à l'information publiée¹⁰. Ces termes constituent ainsi une *expansion de la sémantique* de l'information publiée, et vont permettre d'élargir le champ des réponses lors de sa recherche¹¹. Dans le cas d'un service, un tel élargissement de la sémantique d'un service lors de sa publication n'est pas envisageable. Il y a pour plusieurs raisons à cela, telles que le risque de dilution du but du service, la confusion avec d'éventuels synonymes et homonymes, ou le mélange entre les caractéristiques fonctionnelles et non-fonctionnelles d'un service. Comment faire dans ce cas pour augmenter les chances qu'un service métier proposé dans l'annuaire soit trouvé lors d'une recherche ?
- Enfin, le troisième problème est lié à **la nature même d'un service métier**. En effet, la granularité disponible pour la publication a aussi un impact important sur le niveau d'expression du service. Comme le modèle MiS propose l'agrégation de services, comment faire pour que la description d'un service agrégé tienne compte des services composants et de leurs buts respectifs ?

Notre proposition

Dans notre approche, l'input d'un fournisseur lors de la publication d'un service s'exprime directement en termes métiers sous forme de buts. Le processus de publication exploite la structure linguistique du but (pour construire une description du service à base d'annotations sémantiques. Nous introduisons le guidage dans ce processus pour d'une part, assister le concepteur dans cette tâche, et d'autre part garantir la complétude de la description du service intentionnel publié. Ce guidage est possible grâce à la modélisation du processus de publication sous forme de carte intentionnelle. Dans cette carte, des directives de sélection assistent l'utilisateur dans les multiples choix qui se présentent à lui lors de la construction du descripteur intentionnel du service.

Enfin, notre proposition prend en compte au niveau du descripteur et du processus de publication, les agrégations de services intentionnels telles que définies dans le modèle de MiS (cf. chemin simple, multi-chemin, paquet et segment).

¹⁰ Par exemple, un site d'information sur la ville de Paris va associer avec les mots clés qui le décrivent non seulement les termes "Paris", "Ile de France" mais aussi "capitale", "tourisme", "voyage", ou même "tour Eiffel" et "Champs Élysées".

¹¹ Ce qui permettrait dans notre exemple de retrouver le site de la ville de Paris en faisant une recherche à partir du terme "tour Eiffel" ou "tourisme".

2.2. L'expression au niveau métier des besoins du client

Le client est un utilisateur qui, à partir d'un besoin métier exprimé sous une forme particulière (mots clés, phrases en langage naturel, modèle informel de processus, etc.), pose une requête et cherche à découvrir un ou plusieurs services qui satisfont son besoin. Cette question de la recherche de service est un aspect essentiel de l'approche SOA, c'est une facette spécifique de celle-ci qui se distingue de la facette relative à la publication d'un service [Ecke et Wieringa, 2003], [Papazoglou et Van den Heuvel, 2006]. La recherche d'un service intentionnel soulève en fait plusieurs questions :

- ❑ **La forme de l'input client** : L'utilisateur étant un agent métier, sa description du service recherché est généralement directement liée à son contexte métier. Il souhaite évidemment que cette expression préserve la forme et le contenu de ce contexte. Lors d'une recherche, comment faire pour rapprocher la forme de l'input client de son contexte métier ?
- ❑ **Le processus de formulation de la requête** : Quelle que soit la forme avec laquelle la requête est exprimée, l'utilisateur final doit convertir son besoin métier vers cette forme. Ce processus n'est généralement pas pris en charge par la plateforme de recherche et de publication, et l'utilisateur lui-même fait cet effort de traduction de l'expression métier vers le niveau conceptuel ou technique. Comment introduire une assistance de haut niveau pour aider l'utilisateur final dans la formulation de sa requête ?
- ❑ **Appariement et calcul de similarité** : L'appariement est la mise en correspondance entre les éléments de la requête et ceux des descripteurs des services disponibles dans l'annuaire. Il repose sur le calcul de similarité entre certains éléments de la requête et ceux du descripteur. La nature du descripteur et la forme de l'input client impactent les techniques de calcul de similarité et les algorithmes d'appariement utilisés. Selon la nature des informations stockées dans le descripteur du service, et selon le niveau d'abstraction et de sophistication de l'input client, comment faire pour que les techniques de calcul de similarité et que le processus d'appariement prennent en compte la sémantique sous-jacente au descripteur et à l'input client ?
- ❑ **Mode recherche et mode exploration** : Les approches traditionnelles de recherche de services sont construites d'une manière qui permet de répondre à une requête par les services les plus adaptés. Cette recherche se termine par le résultat trouvé et sans plus de possibilité d'explorer la base existante de services selon des liens sémantiques possibles, tels que l'appartenance au même domaine d'application, synonymie du nom de service, liens de généralisation/spécialisation entre noms de services, etc. Comment faire pour exploiter les possibilités offertes par ces liens afin de proposer d'autres services que l'utilisateur n'a pas envisagés ?
- ❑ **Reformulation (ou expansion) de la requête** : La reformulation est un moyen d'aider l'utilisateur à découvrir de nouveaux services sémantiquement proches et/ou liés à sa requête initiale. Comme indiqué dans un exemple simple précédent, le web classique utilise le « tagging » multiple pour élargir le champ sémantique d'une page web lors de sa publication. Dans le cadre de recherche de web service, comment faire pour élargir le champ de la requête et proposer des reformulations de celle-ci qui augmentent les chances de trouver un service qui correspond au besoin de l'utilisateur ?

Notre proposition

Dans notre approche, l'utilisateur exprime sa requête directement sous la forme d'un but métier. Nous exploitons la structure linguistique du but ainsi qu'une classification puissante des verbes (fournie par une ontologie générique des verbes) et la connaissance spécifique de domaine (fournie par une ontologie de domaine) pour élargir le champ sémantique de la requête. Nous exploitons aussi la structure des services intentionnels composites et à variation (cf. chemin simple, multi-chemin, etc.), pour donner à l'utilisateur la possibilité de parcourir l'arborescence de services agrégés et découvrir ainsi de nouveaux services. Ce raisonnement sur le but initialement exprimé dans la requête permet ainsi de découvrir des services dont les buts sont proches de celui recherché.

Concernant l'appariement, il se fait entre les concepts figurant dans la requête et ceux associés aux descripteurs des services. Il utilise une fonction de calcul de distance basée sur les liens sémantiques entre concepts dans l'ontologie de domaine (cf. synonymie, généralisation, spécialisation).

Le guidage est introduit dans le processus de recherche grâce à une modélisation intentionnelle sous forme de carte. Dans cette carte, des directives de sélection assistent l'utilisateur dans les multiples choix qui se présentent à lui lors de la (re)formulation de la requête et l'exploration de l'espace des services intentionnels disponibles. L'utilisateur est ainsi guidé et assisté pour tirer profit de toutes les possibilités qu'offre le processus de recherche.

2.3. Le besoin d'une sémantique étendue

L'orientation métier des services doit intégrer sous une forme ou une autre une connaissance du domaine. Pour être utile et utilisable, cette connaissance doit être structurée selon le modèle sous-jacent au service. Dans le monde des services web sémantiques, on a mentionné précédemment l'insuffisance des annotations sémantiques de domaine telles que définies dans SAWSDL. L'absence de corrélation entre les annotations et la nature du modèle de service sous-jacent fait que les annotations peuvent référencer plusieurs concepts d'une ontologie de domaine, sans qu'il y ait une spécification du contexte d'usage de ces fragments de connaissance de domaine. L'approche YASA4WSDL est une tentative de réponse à ce problème par l'usage d'une annotation qui combine la connaissance métier (ontologie de domaine) avec la connaissance du modèle de service (ontologie technique), figure 1.5.

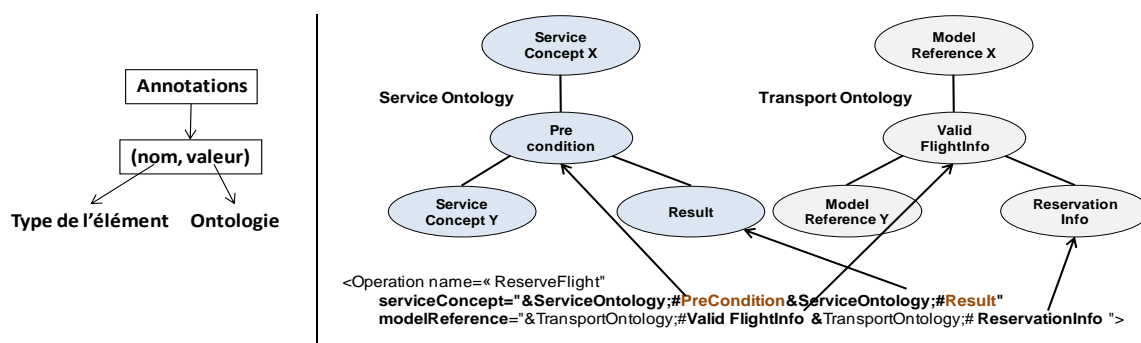


Figure 1.5. Typage d'annotation selon l'approche YASA4WSDL [Chabeb et al., 2009]

Le modèle MiS introduit une sémantique riche dans la représentation d'un service. Cette richesse est due en premier lieu à l'association entre un service et un but métier. Ce but a une structure complexe où la structure linguistique du but contribue à définir la sémantique du

service intentionnel. Elle est due en second lieu aux multiples possibilités qu'offre le modèle pour agréger à de multiples niveaux plusieurs services. Cette structuration des services intentionnels (cf. chemin simple, multi-chemin, paquet, segment) enrichie et complexifie la signification intrinsèque du service intentionnel agrégé. Dans le cadre de notre travail, quelle technique de représentation de la sémantique faut-il introduire pour prendre en compte celle du modèle MiS ?

Notre proposition

Pour accompagner le modèle MiS et l'orientation intentionnelle de sa sémantique sous-jacente, nous adoptons l'approche YASA4WSDL pour la spécification du descripteur d'un service intentionnel. Nous proposons à cet effet l'ontologie technique *iSOnto* pour introduire la connaissance structurée d'un service intentionnel. Les annotations sémantiques utilisées dans le descripteur prennent ainsi en compte la structure du modèle de but (verbe, sujet, paramètres) qui est sous-jacente à MiS.

Enfin, nous accordons une importance décisive à la distinction entre les noms propres (sujet, paramètres) et le verbe dans l'expression d'un but. Grâce à une ontologie générique des verbes¹² et des liens sémantiques entre les verbes et les noms propres, il devient possible de raisonner sur la signification d'un but et d'introduire ainsi intelligence et guidage dans les processus de publication et de recherche de service.

3. APERÇU DE LA SOLUTION

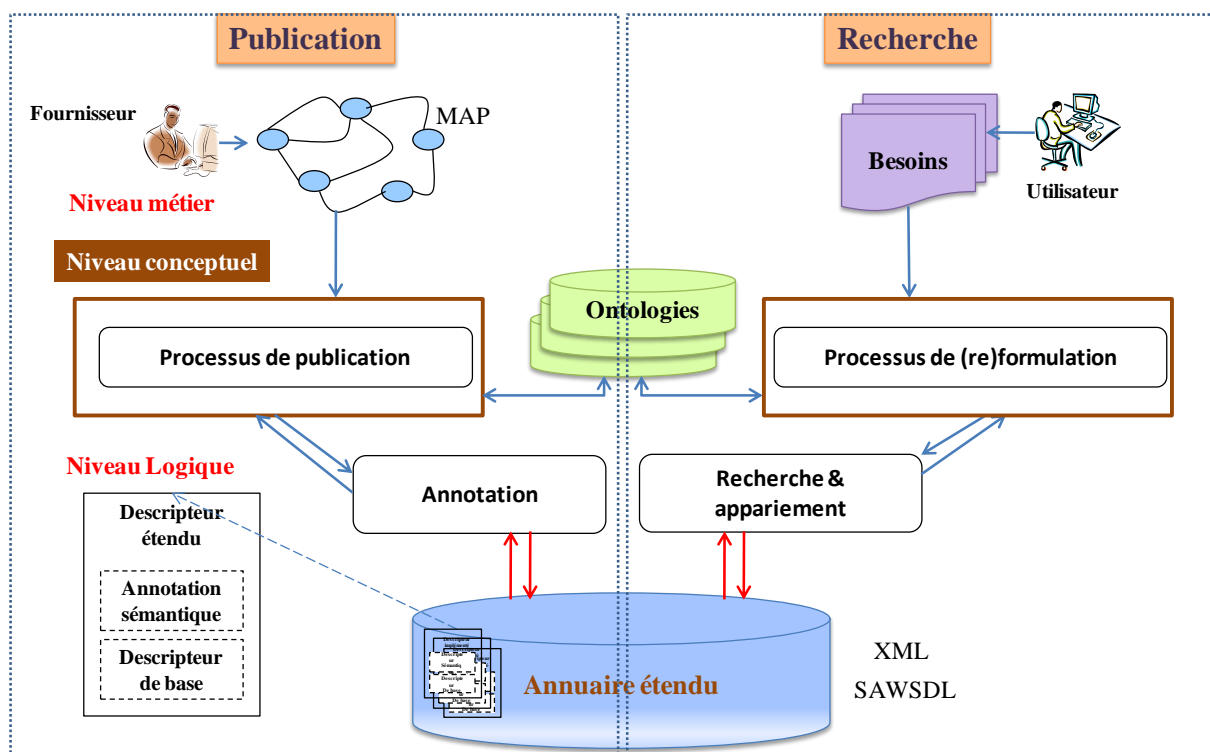


Figure 1.6. Synopsis de la solution

¹² Cette ontologie des verbes est indépendante d'un domaine particulier, elle est directement inspirée de [Urego, 2005]

La figure 1.6 présente une vue générale de la solution proposée. Les principaux objectifs de l'approche *PASIS* sont (i) de guider les fournisseurs dans la publication des services intentionnels de telle manière à préserver la sémantique du service tel qu'il a été initialement identifié par l'approche iSOA, (ii) de permettre aux utilisateurs finaux d'exprimer leurs besoins sous forme de requêtes basées sur le modèle de but, (iii) de fournir une aide intelligente et riche aux utilisateurs qui recherchent les services intentionnels les plus aptes à satisfaire leurs besoins métier, et (iv) de donner la possibilité aux utilisateurs d'explorer l'espace des services intentionnels disponibles dans l'annuaire étendu grâce à la prise en compte de l'agrégation complexe des services.

3.1. Hypothèses

Dans le but de valider l'approche *PASIS* proposée, nous formulons les trois hypothèses suivantes :

Hypothèse 1 :

La modélisation du processus de publication de services intentionnels aboutit à un guidage utile pour le fournisseur de service.

Hypothèse 2 :

La modélisation du processus de requête de services intentionnels aboutit à un guidage utile pour l'utilisateur de service.

Hypothèse 3 :

Les liens entre les ontologies de verbes et l'ontologie de produits enrichissent le processus de recherche et améliorent le résultat des requêtes.

Ces trois hypothèses seront testées grâce à une illustration de l'approche *PASIS* sur un exemple complet issu du domaine de la santé.

3.2. Contributions

Les principales contributions du travail de recherche réalisé dans cette thèse sont les suivantes:

- La proposition d'un *processus de méthodologique pour la publication modélisé sous forme de carte*. Ce processus aide le fournisseur dans sa démarche de publication par la complétude et la résolution de conflits dans l'annotation.
- L'introduction d'une *ontologie technique iSOnto* nécessaire pour pouvoir mettre en place l'annotation sémantique étendue.
- La définition d'une *extension de l'annuaire avec un descripteur de service intentionnel annoté sémantiquement*. Ce descripteur utilise des annotations sémantiques étendues pour prendre en compte les intentions de services dans le descripteur standardisé de service.
- L'utilisation du modèle de buts *pour la spécification des besoins métiers sous forme de requêtes*. Ceci apporte une homogénéité conceptuelle dans la perception d'un service intentionnel au sein de l'architecture iSOA.
- La proposition d'un *processus de formulation de requêtes modélisé sous forme de carte*. Ce processus méthodologique aide l'utilisateur dans la formulation de la requête, la recherche des services et l'exploration des services composants et/ou

composites. La reformulation de requête aide à converger vers les services recherchés, alors que l'exploration permet de découvrir d'autres possibilités auxquelles l'utilisateur n'a pas forcément pensé.

- La définition et l'utilisation d'un *algorithme d'appariement* pour mesurer le degré de similarité entre les concepts d'une ontologie. Cet algorithme est utilisé lors de la reformulation d'une requête pour proposer des concepts alternatifs à ceux définis dans la requête de l'utilisateur et augmenter ainsi la possibilité de trouver des services intentionnels proches.
- L'implémentation d'un *prototype de l'approche* montrant le déroulement des deux processus (publication, recherche) basé sur l'extraction des termes et l'application de l'algorithme d'appariement.

4. PLAN DU MEMOIRE

La suite de ce mémoire est composée des chapitres suivants :

Chapitre 2 : *État de l'art*. Ce chapitre présente une revue de la littérature relative à la publication et à la recherche des services. On introduit un cadre de référence pour comparer les différentes approches existantes et positionner la notre.

Chapitre 3 : L'approche *PASIS* (**P**ublishing **A**nd **S**earching of intentional **S**ervices). Ce chapitre décrit d'une manière globale l'approche et ses caractéristiques. Également, il présente la définition et la construction de l'ontologie technique de services intentionnels *iSOnto*, ainsi que l'ontologie des verbes et l'ontologie des produits et la relation entre les deux. Ce qui permet de spécifier le descripteur d'un service intentionnel. Enfin, on présente brièvement le modèle de Carte qui est la base du modèle MiS et qui est utilisé pour modéliser les processus de publication et de recherche.

Chapitre 4 : *La publication de services intentionnels*. Ce chapitre présente le processus méthodologique pour la publication des services intentionnels, et introduit ainsi les différentes stratégies pour guider le fournisseur de services. Ce processus est décrit sous forme de carte intentionnelle avec le formalisme MAP.

Chapitre 5 : *La recherche de services et la reformulation des requêtes*. Ce chapitre commence par définir notre algorithme de calcul du degré de similarité. Ensuite est introduit le processus de formulation des requêtes des utilisateurs et de recherche des services intentionnels. Ce processus est décrit sous forme de carte intentionnelle avec le formalisme MAP.

Chapitre 6 : *Cas d'application*. Ce chapitre illustre notre approche par une étude de cas. Cette étude de cas est issue du domaine de santé, elle concerne la gestion des activités d'une clinique à travers la réception d'un patient, le diagnostic d'une maladie, la prescription d'un traitement et la facturation des actes. Ce cas illustratif apporte des éléments de validation aux hypothèses formulées précédemment (cf. 3.1) et permet de discuter la pertinence de notre approche. Ce chapitre se termine par la présentation d'un prototype en cours de construction autour de l'approche *PASIS*.

Chapitre 7 : *Conclusion et perspectives*. Ce chapitre conclut la thèse et propose des perspectives de recherche faisant suite à ce travail.

CHAPITRE 2

ETAT DE L'ART

1. INTRODUCTION

Ce chapitre présente un état de l'art des travaux proposés dans la littérature relatifs à l'ingénierie de SI à base de services. Il traite en particulier des différents modèles, langages et systèmes pour la description, la publication, la formulation de requêtes et la recherche de services, en accordant une importance particulière à celles qui sont dirigées par les buts.

Pour présenter et mettre en perspective ces différentes approches, nous introduisons un cadre de référence inspiré du « cadre des quatre vues » [Rolland *et al.*, 1998]. Ce cadre fournit une structure de caractérisation selon quatre perspectives des approches étudiées qui facilite leur comparaison. Chaque perspective correspond à une dimension et à un aspect essentiel de l'approche étudiée. Ce cadre permet ainsi (1) d'identifier des problèmes impliquant les approches d'ingénierie à base de services et (2) de positionner ces approches les unes par rapport aux autres. La structure du cadre de référence a été obtenue par analyse des sources d'information étudiées. Cette structure permet de synthétiser les forces et faiblesses de chacune des 7 approches de publication et de recherche de services par rapport aux caractéristiques identifiées dans chacune des vues. Ces approches ont été choisies parce qu'elles apportent un point de vue original au problème de recherche de services.

La section 1 décrit le cadre de référence spécifique à la publication et à la recherche de services. Ce cadre est ensuite utilisé, dans la section 2 pour présenter les différentes approches choisies. Dans la section 3, un tableau récapitulatif synthétise l'analyse effectuée et permet de mettre en évidence et de discuter les principales caractéristiques des approches étudiées. Enfin, la section 4 conclue ce chapitre et introduit le positionnement par rapport à l'état de l'art de l'approche proposée dans cette thèse.

1. CADRE DE REFERENCE POUR LES SERVICES

Les quatre vues du cadre de référence [Rolland *et al.*, 1998], ont prouvé leur efficacité dans l'amélioration de la compréhension des différentes disciplines de l'ingénierie, telles que

l'ingénierie des systèmes d'informations [Jarke *et al.*, 1992], l'ingénierie des exigences [Jarke et Pohl, 1993], l'ingénierie des processus de développement des systèmes d'information [Rolland *et al.*, 1998], l'ingénierie des méthodes [Rolland, 1997] et l'ingénierie des systèmes à base de services [Kaabi, 2007]. Nous considérons qu'un cadre similaire peut être utilisé pour aider à comprendre les approches de publication et de recherche de services en tenant compte des techniques, des outils et des méthodes utilisés. Le but de ce cadre est donc d'être un outil intellectuel et conceptuel pour analyser l'état de l'art des approches étudiées et de mettre en évidence les points essentiels à analyser et à comparer.

1.1. Méta-modèle utilisé

Ce cadre de référence est complété par des vues selon l'approche introduite dans [Prieto-Diaz, 1991]. Celle-ci vise une classification flexible et précise des composants logiciels et s'appuie sur l'énumération des descripteurs des composants, leur association à un lexique de termes (thésaurus) et à un graphe des vues. Le cadre initial des quatre vues est adaptable par la définition des facettes constituant le descripteur associée à une vue pour l'approche à caractériser.

Chaque vue est caractérisée par un nom et mesurée à l'aide d'un ensemble de facettes. Toutes les facettes possèdent des valeurs définies dans un domaine. Il existe trois types de domaine : le type simple, le type énuméré et le type ensemble. Le type simple correspond aux types prédéfinis tel que entier, booléen, string. Le type énuméré (noté Enum {valeur x, valeur y}) permet d'exprimer que la facette aura comme valeur une des valeurs de la liste définie à l'aide du type Enum. Enfin, lorsqu'une facette peut prendre une ou plusieurs valeurs d'une liste prédéfinie, alors le type « Ensemble » est utilisé.

Chaque vue du cadre de référence est constituée d'un ensemble de facettes, chacune de ces facettes est définie par un nom et est associée à un domaine de valeur. Le tableau 2.1 introduit un exemple de chacun des trois types de domaines (simple, énuméré et ensemble) avec un exemple de valeur conforme à son type.

Tableau 2.1. Exemple de vue Objet avec ses facettes, domaines et exemples de valeurs

Facette	Domaine	Exemple de valeur
Facette 1	Entier	1
Facette 2	Ensemble (valeur 1, valeur 2, valeur 3)	Valeur 1, valeur 3
Facette 3	Enum {valeur X, valeur Y, valeur Z}	Valeur Z

Le méta-modèle de la Figure 2.1 définit le cadre proposé.

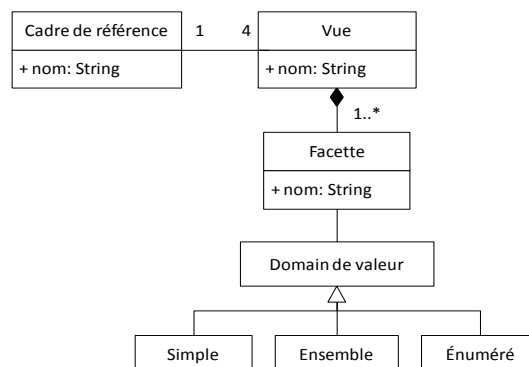


Figure 2.1. Méta-modèle du cadre de référence

1.2. Description du cadre de référence

Comme le montre la Figure 2.2, le cadre de référence que nous proposons ici est obtenu par instanciation du méta-modèle de la figure 2.1 et s'articule autour de quatre vues (sujet, usage, modèle et implémentation).

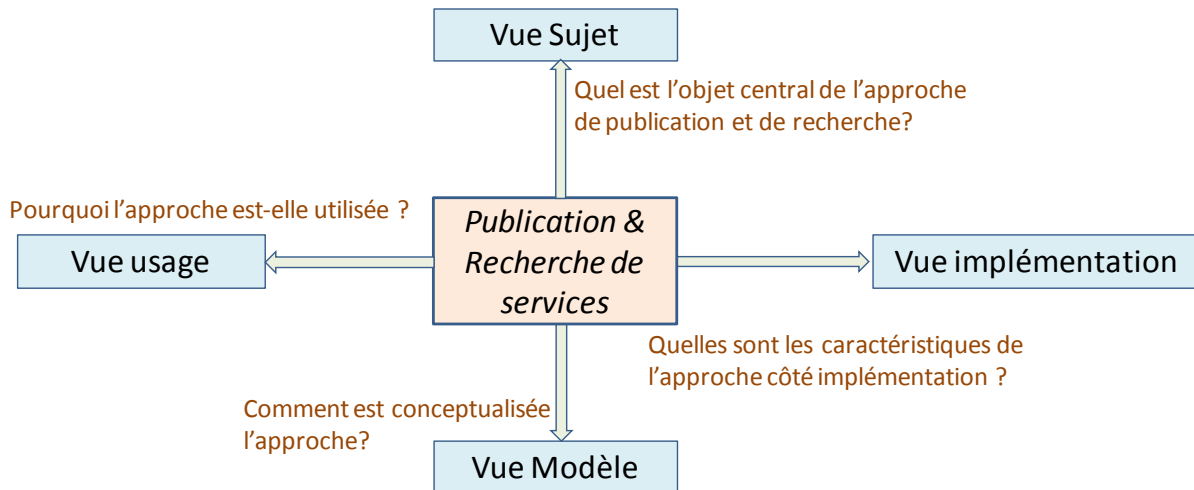


Figure 2.2. Les quatre vues du cadre de référence

Chaque vue permet d'analyser un aspect particulier en posant une question fondamentale. Les quatre questions posées sont celles du "quoi", du "pourquoi", du "comment", et enfin du "par quel moyen". Elles permettent de discuter d'une manière ciblée les différents problèmes à travers les quatre vues :

- (1) la vue « sujet » se centre sur l'objet sur lequel on applique la publication et la recherche de services, c'est-à-dire, l'entité que l'on publie et que l'on recherche, sa granularité et sa visibilité.
- (2) la vue « usage » est le contexte d'usage de l'approche de publication et de recherche de services, elle concerne la finalité et les objectifs de l'approche, les fonctionnalités proposées et la phase du processus d'ingénierie qui est couverte.
- (3) la vue « modèle » concerne les modèles utilisés dans l'approche autant pour les aspects processus que les aspects produits.
- (4) la vue « implémentation » concerne les moyens techniques mis en œuvre pour réaliser l'approche étudiée.

La définition des facettes caractérisant chacune des vues est détaillée dans la suite de cette section.

1.3. La vue sujet

La littérature scientifique traitant des services ou des e-services est très hétérogène. Elle se caractérise par une absence d'unification et d'intégration des concepts rendant difficile une appréhension globale et synthétique de ce domaine. Ce phénomène est accentué par la diversité des visions proposées par les différentes communautés de recherche. Nous proposons de caractériser la vue « sujet » par les trois facettes suivantes : (i) *l'entité*, (ii) *la granularité* et (iii) *la visibilité*. Nous détaillons chacune de ces facettes dans ce qui suit.

1.3.1. L'entité

[Baida et *al.*, 2004] suggère qu'au moins deux perspectives sur les services doivent être comprises afin de fournir une terminologie partagée pour les services : (i) Un point de vue technologique (informatique), (ii) un point de vue métier (business). A ces deux perspectives, nous ajoutons le point de vue *orienté but* qui traduit le fait qu'un service exhibe une intentionnalité formulée par le *but* qu'il permet à ses clients d'atteindre [Rolland et Kaabi, 2005]. Ces différents points de vue sont détaillés dans ce qui suit.

La perspective technologique : La définition d'un service est fortement influencée par les domaines proches de l'ingénierie du web et le développement de standards liés aux services web. Toutefois, dans ce domaine, le terme service est utilisé comme synonyme de web service ou e-service. [Papazoglou et Grefen, 2002] définit un service comme un ensemble d'applications modulaires auto-contenues et auto-descriptives qui peuvent être publiées, localisées et invoquées depuis le web. Un service peut ainsi être vu comme une entité indépendante capable de fournir une certaine fonctionnalité à travers une interface bien définie, laquelle permet son invocation de manière standard, sans que les clients soient obligés de connaître les détails de son implémentation [Issarny et *al.*, 2007] [Sassen et Macmillan, 2005].

La perspective métier : Elle se concentre sur la spécification des besoins des utilisateurs qui sont satisfaits par des services. Au sein de cette perspective, les experts métiers n'ont pas à traiter les détails techniques de la plateforme d'exécution. Selon [Pallos, 2001], un service est un groupement logique de composants requis pour satisfaire une demande métier particulière. Par conséquent, il est intéressant de mutualiser un ensemble de fonctions techniques indispensables aux applications réparties sur le web sous forme de services indépendants et standards, il peut être également avantageux de partager des services ayant trait à certaines grandes fonctions de l'entreprise, quel que soit son secteur d'activité. Ces fonctions sont regroupées dans un service métier. Elles visent finalement à reproduire dans le monde virtuel les transactions commerciales du monde réel (transactions, contrats, facturation, paiement, etc.).

La perspective orientée but : Elle propose de décrire le service métier ou technique par rapport à l'intention ou au but qu'il permet d'atteindre dans un contexte donné. Selon la granularité de l'intention, la réalisation de celle-ci peut se faire par l'exécution d'un service applicatif ou par la composition d'autres services dirigée par les buts. L'avantage de ce modèle est qu'il offre une définition orientée vers les objectifs métiers des utilisateurs. Dans cette perspective orientée but, nous pouvons citer l'approche proposée par [Penserini et *al.*, 2007], et dont la méthodologie de développement *Tropos* [Bresciani et *al.*, 2004] se base aussi sur la notion de but. Cette approche adopte un framework de conception pour modéliser et analyser les besoins initiaux dans l'ingénierie logicielle. Le framework se concentre sur la capture et l'analyse des utilisateurs et de leurs buts, afin de guider la conception du futur système. Cette conception est composée d'agents logiciels qui ont leurs propres buts et leurs propres capacités et qui sont destinés à supporter la réalisation des buts client.

Pour résumer, l'entité mise en jeu peut donc être de trois types différents :

Entité : Ensemble {technologique, métier, orientée but}

1.3.2. Granularité

Un service Web peut être soit *atomique* ou bien *agrégat*. [Fauvet et *al.*, 2002] introduisent une approche pour modéliser la composition des services, dans laquelle un service agrégat est

défini comme un regroupement des services (agrégats ou atomiques). Dans la vision iSOA [Rolland *et al.* 2008], les auteurs introduisent la notion de granularité de service intentionnel. Un service agrégat est composé d'autres services qui peuvent être atomiques ou eux-mêmes des services agrégats. Il y a donc une composition récursive des services. La décomposition est de type ET/OU et se traduit par l'introduction dans le méta-modèle MiS [Kaabi, 2007] par deux types de services agrégats, les services à variation et les services composites. Les premiers établissent des liens OU entre services composants et offrent des choix dans la manière d'atteindre le but du service ; les seconds établissent des liens ET entre services composants. Un service peut être atomique ou agrégat, et un service agrégat peut être composite ou à variation.

La facette de la granularité est définie comme suit :

Granularité : Ensemble {atomique, agrégat}
--

1.3.3. Visibilité

Selon la définition de [Arni-Bloch *et al.* 2009], on distingue deux types de spécification de services : (i) les services décrits comme une *boîte noire*, avec une partie interface unique disponible à d'autres services lui permettant d'être composé ; et (ii) les services décrits comme une *boîte blanche*, lesquels incluent dans leur spécification les informations sur les règles, les processus et la structure du service, informations qui sont donc partagés avec d'autres services.

De manière similaire, selon la définition d'un service proposée dans [Quartel *et al.* 2004a] « *a regularly interacting or independent group of items forming a unified whole* », on peut distinguer deux perspectives du service : externe référencée par « *unified whole* » et interne référencée par « *interacting or independent group of items* ». D'une part, la perspective externe tient à présenter le service comme une *boîte noire* : seules les entrées/sorties (les messages) sont gérées, sans que les spécifications internes soient fournies. Ceci permet un haut niveau de modularité et d'interopérabilité. L'avantage de ce modèle de message est qu'il permet de s'abstraire de l'architecture, du langage ou encore de la plateforme qui prendra en charge le service. Il suffit juste que le message respecte une structure donnée pour qu'il puisse être utilisé. Cela permet un couplage faible au travers d'un échange asynchrone de messages. D'autre part, la perspective interne tient à présenter le service comme une *boîte blanche*. Le service dans ce cas n'est plus une entité monolithique, mais il est composé d'un groupe d'éléments qui interagissent afin de réaliser la fonction du service.

Le service mis en jeu peut être de deux types comme le montre la définition ci-dessous :

Visibilité : Ensemble {Boîte noire, Boîte blanche}
--

En conclusion la vue *sujet* est définie comme suit :

Tableau 2.2. Liste des facettes de la vue du sujet

Facette	Domaine de valeur
Entité	Ensemble {technologique, métier, orientée but}
Granularité	Ensemble {atomique, agrégat}
Visibilité	Ensemble {boîte noire, boîte blanche}

1.4. La vue usage

Cette facette répond à la question « Pourquoi et comment l'approche est-elle utilisée ? ». Nous proposons de caractériser la vue « usage » par les trois facettes suivantes : (i) *les finalités auxquelles répond l'approche*, (ii) *les fonctionnalités proposées* et (iii) *la phase du processus d'ingénierie qui est concerné par l'approche*.

1.4.1. Finalités

La finalité d'une action en est le « pourquoi », le sens, par opposition à son « comment », au mécanisme ou fonctionnement qu'elle met en jeu. Nous considérons que les finalités d'une approche de services sont centrées autour de quatre points : (i) *Découvrir*, (ii) *Invoquer*, (iii) *Composer* et (iv) *Explorer*.

Découvrir les services : Cet objectif désigne l'identification dans un annuaire de services qui répondent directement ou indirectement aux besoins de l'utilisateur. Ces services auront été préalablement modélisés et publiés dans un annuaire de services. La découverte de service se focalise ainsi sur la phase d'appariement sémantique entre l'offre et la demande de service. On souligne ainsi si l'objectif d'une approche de services comprend la découverte de services.

Invoquer les services : Cet objectif désigne l'appel à distance et l'exécution d'un service logiciel. Comme le client peut n'avoir aucune connaissance détaillée du service qu'il va invoquer (visibilité de type boîte noire), l'invocation repose principalement sur l'utilisation de l'interface du service. Nous voulons souligner ainsi si une approche de services permet explicitement l'invocation de services.

Composer les services : Cet objectif désigne la possibilité donnée à l'utilisateur de construire de nouveaux services à partir d'autres services préexistants. Le nouveau service obtenu est appelé service composite. Un service composite est donc un ensemble organisé d'autres services. Dans ce cas, les services composants sont réutilisés par le service composite. On souligne ainsi si une approche de services permet explicitement la composition de services.

Explorer les services : L'exploration des services consiste à naviguer dans l'ensemble ou un sous-ensemble des services composants ou composites disponibles dans l'annuaire ou trouvés dans la phase de recherche. On souligne ainsi si une approche de services permet explicitement l'exploration de services.

La facette *finalité* de la vue *usage* est définie comme suit :

Finalité : Ensemble {Découvrir, Invoquer, Composer, Explorer}

1.4.2. Fonctionnalités

Une fonctionnalité correspond à une action explicitement proposée par l'approche et qui permet de réaliser une fonction spécifique. Cette facette répond à la question du « comment ». Nous considérons que l'utilisation des approches de services proposent essentiellement cinq catégories d'actions : (i) *Décrire les services*, (ii) *Publier les services*, (iii) *Formuler les requêtes de l'utilisateur à la recherche de services*, (iv) *Rechercher les services* et (v) *Classer les services*.

Décrire les services : La description des services désigne l'opération par laquelle le fournisseur construit la description détaillée du service (input, output, liste d'opérations, ... etc.) en utilisant un langage technique (tel que WSDL) ou un langage de haut niveau (langage

naturel, réseau de Petri, modèle de tâche, etc.). On souligne ainsi si l'approche étudiée propose explicitement la fonctionnalité de description de services.

Publier les services : La publication des services désigne l'opération par laquelle le fournisseur introduit le descripteur du service dans l'annuaire et y associe un certain nombre d'éléments d'information pour faire connaître le service et qui permettent aux utilisateurs de le retrouver par la suite. Cette fonctionnalité de publication peut se limiter à une simple insertion dans l'annuaire et une association à une collection de mots-clés. Elle peut aussi être une opération complexe d'identification sémantique de la nature et du contenu du service et une association avec des concepts dans une ou plusieurs ontologies. On souligne ainsi si l'approche étudiée propose explicitement l'opération de publication des services.

Formuler les requêtes de l'utilisateur à la recherche de services : La formulation des requêtes des utilisateurs désigne l'opération par laquelle l'utilisateur d'un service formule son besoin. Cette requête du client peut prendre plusieurs formes tel qu'un mécanisme spécifique d'interrogation, une formulation directe dans le langage de description de service ou dans un langage technique d'interrogation ; ou bien une expression avec un langage de haut niveau autre que le langage de description de service. On souligne ainsi si l'approche étudiée propose explicitement l'opération de formulation des requêtes.

Rechercher des services : La recherche des services désigne les opérations par lesquelles l'approche retrouve dans l'annuaire les services qui correspondent à la requête formulée par l'utilisateur. La recherche de service fait généralement appel à des techniques d'appariement (lexicale, sémantiques, ontologiques, etc.) entre l'offre et la demande de service. Cette opération de mise en correspondance ou de « mappings » utilise une similarité relative selon une fonction de mesure donnée [Klein et al., 2005]. On souligne ainsi si l'approche étudiée propose explicitement l'opération de recherche de services.

Classer les services : La classification des services désigne l'opération qui permet de trier les services trouvés selon certains critères. En général, dans un scénario réel, étant donné une demande de service, il est concevable qu'il existe des dizaines de services qui répondent à des degrés divers à la requête de l'utilisateur. Par conséquent, il est très important de préciser certains critères de classement. L'approche traditionnelle pour le classement des résultats de *matching* est généralement basée sur le degré de correspondance que fournit la fonction de mesure de similarité [Li et Horrocks, 2003] [Paolucci et al., 2002a]. On souligne ainsi si une approche propose explicitement la classification de services en prenant en compte les techniques et les algorithmes d'appariement.

La facette *fonctionnalité* de la vue *usage* est définie comme suit :

Fonctionnalité : Ensemble {Décrire, Publier, Formuler, Rechercher, Classer}

1.4.3. Phase

Il s'agit de la phase du processus d'ingénierie à base de service qui est concernée et couverte par l'approche. Nous considérons que la phase peut prendre deux valeurs : (i) *Design time* et (ii) *Run time*.

Design time : Le *design time* est l'étape pendant laquelle l'utilisateur conçoit son système dans une approche d'ingénierie orientée service. A cette phase du processus correspondent généralement des objectifs de découverte et d'exploration de services, ainsi que des fonctionnalités de description de service, de publication, de formulation de requête et de recherche.

Run time : Au contraire de design time, la phase de *Run time* indique l'étape pendant laquelle le système à base de service conçu par l'utilisateur est exécuté dans une perspective de prototypage ou de système final. A cette phase du processus correspondent généralement des objectifs d'invocation et de composition de service.

La facette *phase* de la vue *usage* est définie comme suit :

Phase : Ensemble {Design time, Run time}
--

En conclusion la vue « *usage* » est défini comme suit :

Tableau 2.3. Liste des facettes de la vue de l'usage

Facette	Domaine de valeur
Finalités	Ensemble {Découvrir, Invoquer, Composer, Explorer}
Fonctionnalités	Ensemble {Décrire, Publier, Formuler, Rechercher, Classer}
Phase	Ensemble {Design time, Run time}

1.5. La vue modèles

La vue modèle concerne les concepts, les formalismes et les notations utilisés dans l'approche étudiée. Nous avons donc identifié deux catégories de modèles : (i) le *modèle produit*, et (ii) le *modèle de processus*.

1.5.1. Modèle de produit

Le modèle de produit concerne les modèles utilisés pour décrire le(s) différents produit(s) manipulées dans l'approche étudiée et réaliser les fonctionnalités proposées. Nous avons identifié cinq facettes : (i) modèle *input utilisateur*, (ii) modèle de *requête*, (iii) modèle du *descripteur*, (iv) modèle *input fournisseur* et (v) *nature d'ontologie*. Chacune de ces cinq facettes est détaillée dans l'une des cinq sous-sections ci-dessous.

1.5.1.1. Modèle input utilisateur :

Dans cette facette, nous nous intéressons aux langages qui permettent aux utilisateurs d'exprimer et éventuellement de modéliser leurs requêtes lors de la recherche de services. Nous avons pu identifier quelques formes :

Mots-clés : C'est la forme utilisée pour saisir une requête dans les annuaires standards de service (tels que les annuaires UDDI).

Modèle de tâche : Dans cette forme, l'utilisateur exprime sa requête comme un modèle de tâche métier. Une tâche est une action réalisée par un ou plusieurs agents afin d'atteindre un but. C'est le choix fait dans l'approche proposée par [DaSilva et al., 2009] par exemple.

Modèle de but : Dans cette forme, l'utilisateur exprime sa requête comme un objectif métier exprimée selon une certaine structure. Dans les approches de [Mirbel et Crescenzo, 2009] et [Driss et al., 2010], le formalisme de but utilisé est celui proposé initialement par [Prat, 1997], [Prat, 1999], où l'on distingue le sujet, le verbe et les paramètres.

Modèle de carte : La *Carte* est un système de représentation qui permet de modéliser les processus dans des termes intentionnels [Rolland et al., 1999] [Rolland, 2007]. Le graphe

montre quelles intentions peuvent être réalisées au moyen de quelles stratégies. C'est le choix fait dans l'approche proposée par [Mirbel et Crescenzo, 2009] par exemple.

Pour résumer, La facette *modèle input utilisateur* de la vue *modèle de produit* est définie comme une facette dont la valeur est un texte libre :

Modèle input utilisateur : <Libre> (exemple: *Mots-clés, tâche, but, modèle de la carte, ...*)

1.5.1.2. Modèle de requête

Dans cette facette, on s'intéresse à la forme à partir de laquelle le moteur de recherche associé à l'annuaire effectue la recherche. Cette forme peut correspondre directement à ce que l'utilisateur a formulé dans sa requête, ou bien elle est obtenue par transformation de la requête vers une autre forme que le moteur de recherche peut traiter. Nous avons pu identifier quelques formes :

Mots-clés : C'est le cas où l'input utilisateur sous forme de mots-clés est directement transmis au moteur de recherche, comme dans les annuaires standards UDDI.

XQuery : C'est un langage optimisé pour l'interrogation de tous les types de données XML [XQuery, 2003]. XQuery est une évolution d'un langage de requête appelé Quilt, lui-même basé sur divers autres langages de requête tels que le Xpath (XML Path Language) version 1.0, XQL et SQL. La conception du langage XQuery est basée sur un ensemble d'expressions qui peuvent être composées ensemble arbitrairement. Dans l'approche [Klüh et al., 2006] par exemple, l'input client sous forme de descripteur de service est transformé en une requête XQuery. De même, dans l'approche [Zachos et al., 2007], l'input client sous forme de phrases en langage naturel est transformé en requête XQuery.

SPARQL : Ce langage combine un langage d'interrogation et un protocole pour créer un service web au sens propre [W3C, 2008]. Une spécification SPARQL définit un langage d'interrogation et un protocole qui fonctionnent en parfaite synergie avec les autres technologies du Web sémantique (telles que RDF et OWL). Dans l'approche [Mirbel et Crescenzo, 2009], une requête composée à partir de fragments prédéfinis de modèle de carte est transformée en une requête SPARQL.

Pour résumer, La facette *modèle de requête* de la vue *modèle de produit* est définie comme une facette dont la valeur est un texte libre :

Modèle de requête : <Libre> (exemple: *Mots-clés, XQuery, SPARQL, ...*)

1.5.1.3. Modèle de descripteur

Dans cette facette, on s'intéresse aux modèles et aux langages pour la description d'un service côté fournisseur. La proposition initiale est celle du triplet de standards WSDL, SOAP et UDDI dans le cas des services web simples. Le besoin d'une sémantique étendue a abouti à l'introduction des services web sémantiques. On a donc identifié les valeurs suivantes pour la facette du modèle de descripteur :

WSDL¹³ : la forme de base pour la description d'un service web [Chinnici et al., 2007].

WSMO¹⁴ : Cette approche est similaire à la précédente, WSMO est une ontologie de services basés sur WSMF¹⁵ [Fensel et Bussler, 2002], [Lara et al., 2004], [ESSI WSMO working

¹³ Web Service Description Language

¹⁴ Web Service Modeling Ontology

group, 2004]. WSMO définit quatre éléments de haut niveau suivants pour décrire les services Web sémantique : les ontologies, les services, les médiateurs et les objectifs.

OWL-S¹⁶ : L'approche OWL-S propose une ontologie supérieure pour les services Web motivée par la nécessité de fournir trois types essentiels de connaissance sur un service Web : (i) Qu'est ce que fournit le service aux clients potentiels, (ii) Comment est il utilisé et (iii) Comment peut-on interagir avec le service ? En effet, les parties « *profile* », « *processus* » et « *grounding* » permettent respectivement de répondre aux questions (i), (ii) et (iii).

SAWSDL¹⁷ : [Farrell et Lausen, 2007] suggère comment ajouter des annotations sémantiques pour les diverses parties d'un document WSDL (telles que les interfaces, les opérations, etc.). Cette extension est conforme au cadre d'extensibilité de WSDL. SAWSDL définit ainsi un nouvel espace de nommage appelé "*sawSDL*" et ajoute un attribut d'extension appelé "*modelReference*" afin que les relations entre les composants WSDL et les concepts dans un autre modèle sémantique d'affaires (ontologie, par exemple) soient traités.

SAWSDL étendu : Dans SAWSDL, l'absence de corrélation entre les annotations sémantiques et la nature du modèle de service sous-jacent fait que les annotations peuvent référencer plusieurs concepts d'une ontologie de domaine, sans qu'il y ait une spécification du contexte d'usage de ces fragments de connaissance de domaine. L'extension proposée par l'approche YASA4WSDL est une tentative de réponse à ce problème par l'usage d'une annotation qui combine la connaissance métier (ontologie de domaine) avec la connaissance du modèle de service (ontologie technique).

Pour résumer, La facette *modèle de descripteur* de la vue *modèle de produit* est définie comme une facette dont la valeur est un texte libre :

Modèle de descripteur : <libre> (exemple: *WSDL*, *SAWSDL*, *SAWSDL étendu*, *OWL-S*, *WSMO*, ...)

1.5.1.4. Modèle input fournisseur

Dans le *modèle input fournisseur*, nous nous intéressons aux formes que peuvent prendre les données saisies par le fournisseur lorsqu'il s'agit de décrire et de publier un service. Généralement, ces données prennent la forme du descripteur de service (WSDL, WSMO,...), mais peuvent être plus sophistiquées et orientées vers le contexte métier. Ce modèle est significatif lorsque la fonctionnalité de publication est fournie par l'approche.

Pour résumer, La facette *modèle input fournisseur* de la vue *modèle de produit* est définie comme une facette dont la valeur est un texte libre :

Modèle input fournisseur : <libre>

1.5.1.5. Nature d'ontologie

Dans cette facette, nous nous intéressons à la catégorie de la connaissance qui est représentée dans l'ontologie utilisée par l'approche étudiée. Cette facette est significative lorsque l'approche en question est basée sur les services web sémantiques. Nous avons identifié trois valeurs possibles pour cette facette :

¹⁵ Web Service Modeling Framework

¹⁶ Ontology Web Language for Services

¹⁷ Semantic Annotation Web Service Description Language

Générique : Ces ontologies décrivent des concepts généraux. Elles sont indépendantes d'un domaine d'application particulier, c'est le cas notamment de l'ontologie WordNet et de l'ontologie des verbes proposée par [Urrego, 2005].

Technique : Cette catégorie d'ontologies contient la connaissance sur les concepts qui définissent les éléments utilisés dans un domaine technique tel que celui des services web [Chabeb et al., 2009]. Cette ontologie sert à expliciter la structure des éléments techniques manipulés par l'approche.

Domaine : Cette catégorie d'ontologie décrit les différents objets qui peuvent être utilisés dans un domaine métier particulier tel que la santé ou le tourisme. Elle définit ainsi une classification des éléments de ce domaine.

Pour résumer, La facette *nature d'ontologie* de la vue *modèle de produit* est définie comme un ensemble de valeurs :

Nature d'ontologie : Ensemble {Générique, Technique, Domaine}

1.5.2. Modèle de processus

Le modèle de processus dans une approche d'ingénierie à base de service décrit d'une manière structurée (et éventuellement formelle) la démarche que doit suivre l'utilisateur (client ou fournisseur) de cette approche. Nous avons identifié trois facettes : (i) *processus de publication*, (ii) *processus de formulation*, et (iii) *processus d'appariement*. Chacune de ces trois facettes est détaillée dans l'une des trois sous-sections ci-dessous.

1.5.2.1. Processus de publication

Cette facette concerne la démarche de publication des descripteurs de services. Il n'est significatif que si la fonctionnalité *Publier* est prévue dans l'usage de l'approche. Nous avons identifiés trois valeurs possibles pour caractériser le processus de publication : (i) *brut*, (ii) *guidé* et (iii) *intelligent*.

Brut : Le processus de publication est *brut* lorsque le fournisseur doit manuellement saisir les données pour la description du service. Ce processus est entièrement à sa charge, et aucun support n'est fourni par le système.

Guidé : Le processus de publication est *guidé* lorsque le système dispose d'une certaine connaissance sur la manière avec laquelle ce processus doit être conduit. Cela suppose qu'une certaine modélisation préalable de ce processus a été faite, et que cette modélisation est exploitable par le système. Ce guidage permet de projeter la logique du métier sur les concepts du modèle technique. Il permet également de garantir l'utilisation correcte du modèle du descripteur.

Intelligent : Le processus de publication est *intelligent* lorsque le système est capable de raisonner sur la connaissance qui est véhiculée dans le descripteur du service en cours de publication. Cela suppose que le système repose sur l'utilisation d'ontologies, et que ces ontologies sont utilisées pour raisonner sur les concepts nécessaires aux annotations du descripteur.

Pour résumer, La facette *processus de publication* de la vue *modèle de processus* est définie comme suit :

Processus de publication : Ensemble {Brut, Guidé, Intelligent}

1.5.2.2. Processus de formulation de requête

Cette facette concerne la démarche avec laquelle l'utilisateur final formule et construit une requête. Nous avons identifiés trois valeurs possibles pour caractériser le processus de formulation : (i) *brut*, (ii) *guidé* et (iii) *intelligent*.

Brut : Le processus de formulation est *brut* lorsque l'utilisateur doit manuellement saisir la requête. Ce processus est entièrement à sa charge, et aucun support n'est fourni par le système.

Guidé : Le processus de formulation de requête est *guidé* lorsque le système dispose d'une certaine connaissance sur la manière avec laquelle ce processus doit être conduit. Cela suppose qu'une modélisation préalable de ce processus a été faite, et que cette modélisation est exploitable par le système. Ce guidage facilite la formulation d'une requête et permet d'exploiter toutes les possibilités offertes par le modèle (ou le langage) d'interrogation.

Intelligent : Le processus de formulation de requête est *intelligent* lorsque le système est capable de raisonner sur la connaissance qui est exprimée dans la requête en cours de traitement. Cela suppose que le système repose sur l'utilisation d'ontologies, et que ces ontologies sont utilisées pour raisonner sur les concepts apparaissant dans la requête et ceux référencés dans les descripteurs de services.

Pour résumer, La facette *processus de formulation* de la vue *modèle de processus* est définie comme suit :

Processus de formulation : Ensemble {Brut, Guidé, Intelligent}

1.5.2.3. Processus d'appariement

L'appariement ou le « *matching* » est la mise en correspondance entre les éléments de deux ensembles, en l'occurrence ceux de la requête et ceux des résultats potentiels de cette requête. Ce processus repose sur un ensemble de fonctions permettant de mesurer le degré de similarité entre ces éléments [He et Chang, 2003]. Ceci est réalisé par une ou plusieurs méthodes de comparaison dites « *matcher* » ; ce sont des fonctions utilisées pour calculer la distance (lexicale, sémantique, conceptuelle, grammaticale, etc.) entre ces deux entités. Les « *matchers* » peuvent ainsi combiner plusieurs techniques dans le processus de d'appariement. Nous avons identifiés deux valeurs possibles pour caractériser le processus d'appariement : (i) *Lexical* et (ii) *sémantique*.

Lexical : Dans l'appariement lexical, c'est la forme textuelle et syntaxique des éléments comparés qui est privilégiée [Keller et al., 2005]. La mesure se fait dans ce cas par des métriques connues telles que la mesure Levenshtein, la mesure du vecteur cosinus, la méthode TF/IDF, la méthode de divergence des informations Jensen-Shannon.

Sémantique : Dans l'appariement sémantique, c'est la signification des éléments comparés qui à la base du processus d'appariement. Ce type de processus suppose donc l'existence d'ontologies qui permettent d'associer des hiérarchies de concepts aux éléments de la requête et à ceux des descripteurs de services. La mesure porte donc sur une distance sémantique dans une ontologie.

Pour résumer, La facette *processus d'appariement* de la vue *modèle de processus* est définie comme suit :

Processus d'appariement : Ensemble {Lexical, Sémantique}

En conclusion, la vue *modèle* est définie comme suit :

Tableau 2.4. *Liste des facettes de la vue modèle*

Catégorie d'facette	Facette	Domaine de valeur
<i>Modèle de produit</i>	Input utilisateur	<Libre> (exemple : <i>LN, Tâche, But, Modèle de carte, ...</i>)
	Requête	<Libre> (exemple : <i>Mots clés, SPARQL, XQUERY, ...</i>)
	Descripteur	<Libre> (exemple : <i>WSDL, SAWSDL, SAWSDL étendu, OWL-S, WSMO, ...</i>)
	Input fournisseur	<Libre>
	Nature d'ontologie	Ensemble {Générique, Domaine, Technique}
<i>Modèle de processus</i>	Publication	Ensemble {Brut, Guidé, Intelligent}
	Formulation requête	Ensemble {Brut, Guidé, Intelligent}
	Appariement	Ensemble {Lexicale, Sémantique}

1.6. La vue implémentation

La vue implémentation décrit en général, les techniques, les algorithmes et les architectures logicielles utilisés pour réaliser une approche de publication et de recherche de services. Cette vue comporte trois facettes : (i) la *technique de mesure*, (ii) l'*usage d'ontologie* et (iii) l'*architecture logicielle*. Chacune de ces facettes est détaillée dans les sous sections ci-dessous.

1.6.1. Technique de mesure

Dans La facette *technique de mesure*, nous nous intéressons aux techniques qui permettent de calculer le degré de similarité entre les éléments de la requête de l'utilisateur et ceux des services disponibles dans l'annuaire. Nous avons constaté qu'il existe deux techniques :

Les techniques IR (*Information Retrieval*) : la recherche d'information étudie la manière de répondre pertinemment à une requête en retrouvant de l'information dans un corpus textuel. La mesure de la similarité est basée sur l'appariement lexical et syntaxique. La mesure se fait dans ce cas par des métriques connues telles que la mesure Levenshtein, la mesure du vecteur cosinus, la méthode TF/IDF, la méthode de divergence des informations Jensen-Shannon. Ces techniques sont utilisées dans le processus d'appariement lexical (c.f. 1.5.2.3). Deux notions simples comme le *rappel* (proportion de documents pertinents renvoyés par le système parmi tous ceux qui sont pertinents) et la *précision* (proportion des documents pertinents parmi l'ensemble de ceux renvoyés par le système) ont été étendus pour permettre une analyse fine des performances de système de recherche d'information.

Les techniques sémantiques : ces techniques utilisent les concepts logiques, les règles et les ontologies pour mesurer la similarité entre les concepts. Elles définissent des degrés de similarité sémantique (tels que : *Exact, Subsumes, Subsumed-by, Has-Same-Class, Fail*) pour catégoriser le résultat de la recherche [Li et Horrocks, 2003] [Paolucci et al., 2002a]. Une telle catégorisation fournit également un classement implicite entre les services potentiels.

Pour résumer, La facette *technique de mesure* de la vue *implémentation* est définie comme suit :

Technique de mesure : Ensemble {IR, Sémantique}

1.6.2. Usage d'ontologie

Comme on a pu le voir précédemment dans la description de certaines facettes, les ontologies sont nécessaires lorsque le descripteur du service est sémantique, lorsque le processus de formulation de la requête est intelligent ou lorsque le processus d'appariement est sémantique. Pour mieux caractériser cet usage de l'ontologie dans l'implémentation de l'approche, nous définissons trois valeurs possibles pour cette facette : (i) *Aucun*, (ii) *interne* et (iii) *externe*.

Aucun : L'approche n'utilise pas d'ontologie.

L'usage interne : Les ontologies sont internes par rapport au système de recherche. En effet, ces ontologies sont utilisées dans l'annuaire mais pas exploitées à l'extérieur lorsque les processus de recherche et de publication sont interactifs.

L'usage externe : La possibilité est offerte à l'utilisateur d'interagir directement avec les ontologies pendant le processus de publication et/ou le processus de recherche.

Pour résumer, La facette *usage d'ontologie* de la vue *implémentation* est définie comme suit :

Usage d'ontologie : Enum {Aucun, Interne, Externe}

1.6.3. L'architecture logicielle

Les approches étudiées peuvent donner lieu dans certains cas à des implémentations techniques. Pour cette facette, trois valeurs possibles sont définies : (i) *Aucune*, (ii) *prototype expérimental* et (iii) *outil public*.

Aucune : Aucune implémentation n'a été faite pour l'approche étudiée.

Prototype expérimental : l'approche étudiée a donné lieu à un prototype technique, généralement disponible sous forme de logiciel libre.

Outil public : l'approche étudiée a donné lieu à la construction d'un logiciel disponible publiquement, et qui a été testé et validé, éventuellement par un partenaire industriel.

Pour résumer, La facette *architecture logicielle* de la vue *implémentation* est définie comme suit :

Architecture logicielle : Enum {Aucune, Prototype expérimental, Outil public}

En conclusion la vue *implémentation* est définie comme suit :

Tableau 2.5. Liste des facettes de la vue implémentation

Facette	Domaine de valeur
Technique de mesure	Ensemble {IR, Sémantique}
Usage d'ontologie	Ensemble {Aucune, Externe, Interne}
Architecture logicielle	Enum {Aucune, Prototype expérimental, outil public}

1.7. Résumé du cadre de référence

En synthèse, le cadre de référence montre l'ensemble des apports offerts par les approches de publication et de recherche de services. Ces apports sont décrits dans les quatre vues à savoir sujet, usage, modèles et implémentation. En résumé, les quatre vues du cadre de référence et l'ensemble de facettes de chaque vue sont récapitulés dans la Figure 2.3.

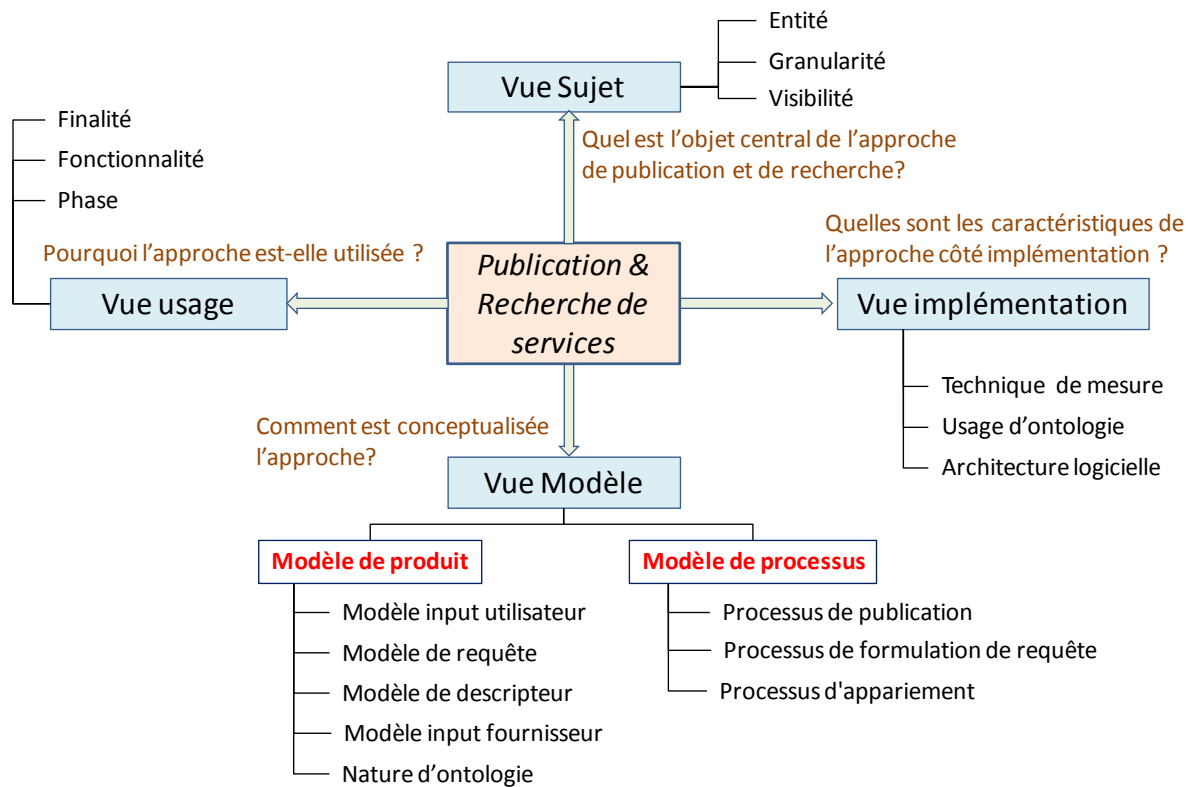


Figure 2.3. Résumé du cadre de référence et les facettes de chaque vue

Les détails des facettes sont récapitulés au tableau 2.6.

Tableau 2.6. *Liste des facettes du cadre de référence*

Vue		Facette	Domaine de valeurs
Sujet		Entité	Ensemble {Technologique, Métier, Orientée but}
		Granularité	Ensemble {Atomique, Agrégat}
		Visibilité	Ensemble {Boite noire, Boite blanche}
Usage		Finalité	Ensemble {Découvrir, Invoquer, Composer, Explorer}
		Fonctionnalité	Ensemble {Décrire, Publier, Formuler, Rechercher, Classer}
		Phase	Ensemble {Design time, Run time}
Modèle	Produit	Input utilisateur	<Libre> (exemple : <i>LN, Tâche, But, Modèle de carte, ...</i>)
		Requête	<Libre> (exemple : <i>Mots clés, SPARQL, XQUERY, ...</i>)
		Descripteur	<Libre> (exemple : <i>WSDL, SAWSDL, SAWSDL étendu, OWL-S, WSMO, ...</i>)
		Input fournisseur	<Libre>
		Nature d'ontologie	Ensemble {Générique, Domaine, Technique}
	Processus	Publication	Ensemble {Brut, Guidé, Intelligent}
		Formulation	Ensemble {Brut, Guidé, Intelligent}
		Appariement	Ensemble {Lexicale, Sémantique}
Implémentation		Technique de mesure	Ensemble {IR, Sémantique}
		Usage d'ontologie	Ensemble {Aucun, Interne, Externe}
		Architecture logicielle	Enum {Aucune, Prototype expérimental, outil publique}

2. POSITIONNEMENT DES APPROCHES AU MOYEN DU CADRE DE REFERENCE

Cette section présente une évaluation de sept approches à base de services :

- (1) Une approche sémantique guidée par les buts pour la recherche de services web à partir des besoins de l'utilisateur : GODO [Gomez et *al.*, 2004] [Gomez et *al.*, 2006].
- (2) Les approches sémantiques pour la recherche hybride de services web : OWLS-MX [Klusch et *al.*, 2006], WSMO-MX [Klusch et *al.*, 2008] et SAWSDL-MX [Klusch et *al.*, 2009].
- (3) Une approche de recherche de services à partir des exigences exprimées en langage naturel structuré [Zachos et *al.*, 2007].
- (4) Une approche pour la composition et la recherche des services web reposant sur les buts des utilisateurs [Da Silva Santos et al. 2008] [Da Silva Santos et al. 2009].
- (5) Une approche sémantique guidée par les intentions pour la recherche de services web : SATIS [Mirbel et Crescenzo, 2009] et [Corby et *al.*, 2009].
- (6) Une approche sémantique pour l'appariement de services web : YASA [Chabeb et Tata, 2008], [Chabeb et *al.*, 2009] et [Chabeb et *al.*, 2010].
- (7) Une Approche centrée-exigence de la modélisation, la recherche et la sélection de services web [Driss et *al.*, 2010].

Ces sept approches ont été sélectionnées parmi toutes celles étudiées dans ce travail parce qu'elles forment un ensemble représentatif de l'état de l'art. En effet celles-ci proposent, d'une part un processus de formulation de requêtes original (intelligent ou guidé) fondé sur le langage naturel ou des modèles sémantiques à base de but, ou d'autre part, d'améliorer significativement (i) les algorithmes d'appariement de services, et (ii) l'extension sémantique exploitée dans les annuaires de services.

Cette partie décrit ces approches et les caractérise selon le cadre de référence présenté précédemment.

2.1. L'approche GODO [Gomez et *al.*, 2006]

GODO [Gomez et *al.*, 2004] [Gomez et *al.*, 2006] est une approche de découverte et d'invocation de services web. L'objectif principal de GODO est d'aider les utilisateurs à formuler leurs besoins pour rechercher de services web sémantique. Il permet aux utilisateurs d'exprimer leurs besoins à travers des textes en langage naturel. Il peut même aider les utilisateurs à formuler leurs besoins grâce à un éditeur intelligent. Pour cela, l'approche GODO fait usage des ontologies sur les actions les plus invoquées qui sont stockées dans un référentiel. Une fois que GODO a analysé l'input de l'utilisateur, il extrait les actions similaires à la demande de l'utilisateur et sélectionne les plus proches en les exprimant sous forme de buts. Le couplage de GODO avec un annuaire sémantique tels que WSMX [Oren, 2004], DAML-S virtuelle machine [Paolucci et *al.*, 2003] et METEOR-S [METEOR-S, 2005] [Patil et *al.*, 2004] permet de réaliser la recherche. Néanmoins, le but n'étant pas intégré dans le descripteur sémantique de ces annuaires, celui-ci est transformé en mots-clés par le *goal sender*.

La figure 2.4 illustre les principales composantes de GODO. La composante essentielle est le *contrôleur* (GODO Control Manager), qui supervise le processus de formulation de la requête. *L'interface graphique* (GUI) permet à l'utilisateur d'exprimer sa requête selon deux possibilités : (i) directement en langage naturel (forme restreinte), ou (ii) assisté par un outil exploitant l'ontologie d'action (Ontology-guided Input) pour l'aider à exprimer ses besoins. Lorsque la requête de l'utilisateur est exprimée, le *contrôleur* délègue à *Goal Loader* la

récupération de tous les buts possibles à partir du *référentiel* (Goal Template Repository) et utilise le *goal matcher* pour déduire les buts proches de la requête. Enfin, le *Goal Sender* transforme les buts composants la requête selon la structure du descripteur sémantique supportée par l'annuaire sélectionné.

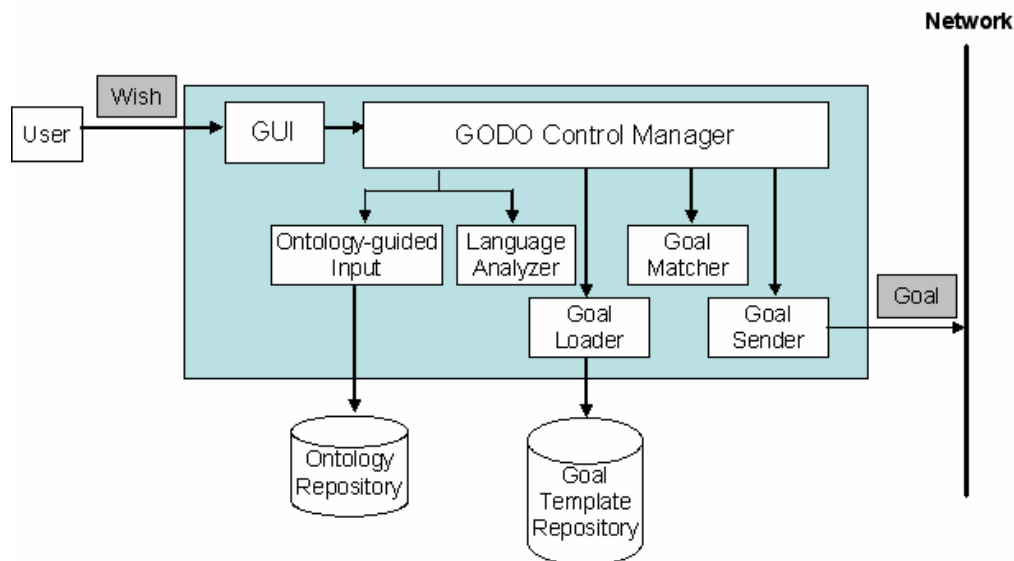


Figure 2.4. L'architecture de GODO

En résumé, l'approche GODO est présentée dans le tableau 2.7.

Tableau 2.7. Les caractéristiques de l'approche GODO [Gomez et al., 2006]

Vue		Facette	Valeurs
Sujet		Entité	Technologique
		Granularité	Atomique
		Visibilité	Boîte noire
Usage		Finalité	Découvrir
		Fonctionnalité	Formuler, Rechercher
		Phase	Design time
Modèle	produit	Input utilisateur	LN (but)
		Requête	Mots clés
		Descripteur	WSML
		Input fournisseur	-
		Nature d'ontologie	Domaine
	processus	Publication	
		Formulation	Intelligent
		Appariement	Lexical

Implémentation	Technique de mesure	?
	Usage d'ontologie	Externe
	Architecture logicielle	Prototype expérimental

L'approche GODO est fondée sur une notion de service atomique de type boîte noire selon une perspective technologique. Néanmoins, le service n'est pas décrit selon une perspective orientée but puisque cette approche est indépendante du descripteur sémantique du service. En effet, la notion de but est utilisée dans GODO uniquement dans le processus de formulation.

L'originalité de cette approche est de proposer un processus de formulation de requêtes intelligent et indépendant de l'annuaire de services sémantiques. GODO prend en compte uniquement la découverte et la recherche de services lors de la conception du système (design time). Le processus de publication et l'algorithme de l'appariement (lexical) sont laissés à la charge de l'annuaire de services. Par contre, les ontologies sont utilisées à l'extérieur de l'annuaire par le processus de formulation.

2.2. L'approche SAWSDL-MX [Klusch et al., 2009]

SAWSDL-MX [Klusch et al., 2009] est une approche de découverte de services web sémantiques. L'objectif principal de SAWSDL-MX est d'aider les utilisateurs à rechercher et classer les services web sémantiques. Pour cela, il fait usage d'ontologies de domaine pour faire un appariement hybride qui utilise une combinaison de mécanismes d'appariement lexical et sémantique. Les auteurs affirment qu'une telle approche, basée sur l'appariement lexical ou sur l'appariement sémantique seulement, échouerait en raison de ses limitations. Une approche hybride, c'est à dire une combinaison de l'appariement lexical et sémantique, donnerait de meilleures performances. En OWLS-MX [Klusch et al., 2006], les auteurs proposent l'appariement sémantique hybride des entrées/sorties dans les « profiles » d'OWL-S. Ils exploitent un raisonnement basé sur la logique et des techniques de recherche d'information basées sur le contenu. Ils proposent cinq degrés d'appariement : les trois premiers sont basés sur la logique uniquement (*exact*, *plug in* et *subsume*), les deux derniers sont hybrides et comprennent des calculs supplémentaires pour mesurer la similarité syntaxiques (*subsumé par* et le *plus proche voisin*). En WSMO-MX [Klusch et al., 2008], le moteur d'appariement accepte comme entrée les services spécifiés en WSML-MX. Les degrés de similarité sont calculés en cumulant les évaluations des quatre éléments correspondants : le type basé ontologie, la contrainte logique, le nom de la relation et la correspondance syntaxique. Les degrés d'appariement sémantique entre un service WSMO et le but (à la demande de service) sont : *équivalence*, *plugin*, *inverse-plugin*, *intersection*, *similitude floue*, *neutre* et *disjonction* [Klusch et al., 2008]. Les métriques de similarité du texte utilisées sont : Perte d'Informations (LI), Jaccard étendu (ExtJacc), cosinus (COS), ou Jensen-Shannon (JS) [Klusch et Kapahnke, 2008].

Figure 2.5 donne un large aperçu de l'architecture globale du système. Fondamentalement, SAWSDL-MX se compose des éléments suivants : *SAWSDL Matching Engine*, *Service Registry*, *Ontology Handlers*, *Local Matchmaker Ontology* et *Similarity Measures*. Du point de vue des fournisseurs de services, SAWSDL-MX permet l'enregistrement des offres SAWSDL service Web à la base de registre. Pour les utilisateurs, SAWSDL-MX fournit une interface pour envoyer des requêtes au moyen d'un document précisant les détails SAWSDL

sur l'interface de service souhaité. Finalement, le processus de recherche de service dépend du moteur de découverte SAWSDL-MX qui retourne une liste triée des offres de service correspondant à la requête.

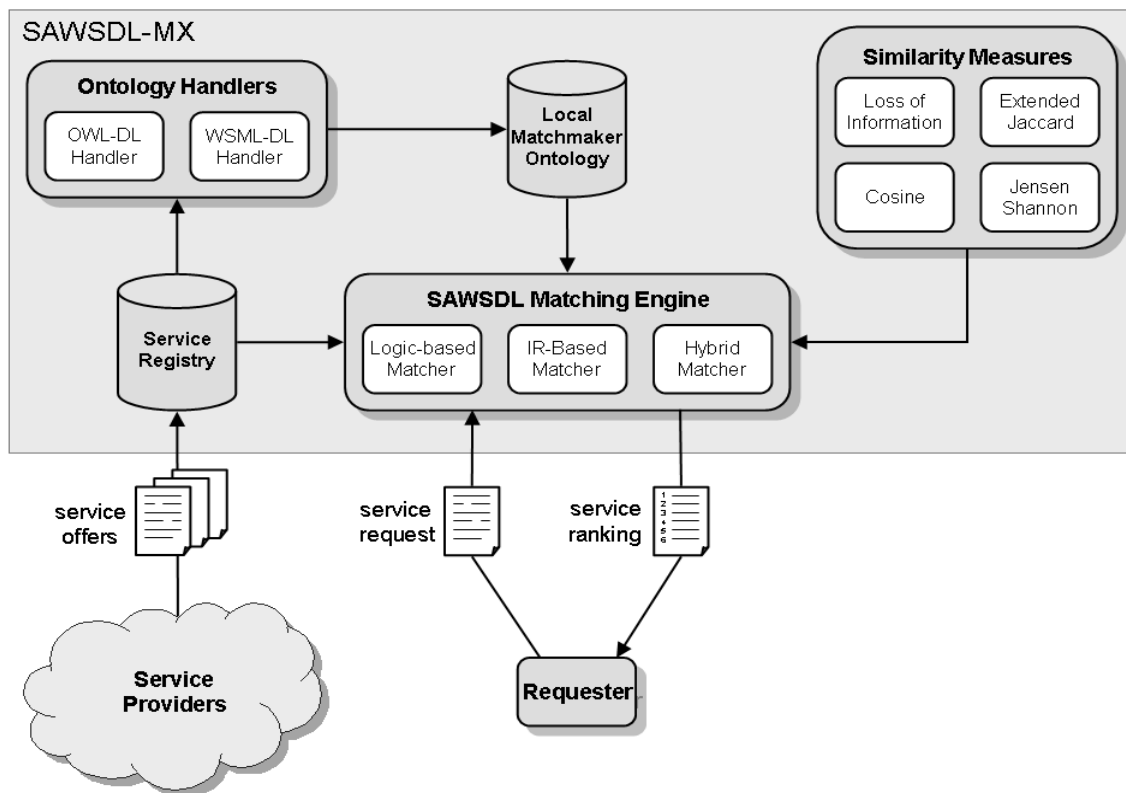


Figure 2.5. L'architecture de SAWSDL-MX

En résumé, l'approche SAWSDL-MX est présentée dans le tableau 2.8.

Tableau 2.8. Les caractéristiques de l'approche SAWSDL-MX [Kluch et al., 2009]

Vue		Facette	Valeurs
Sujet		Entité	Technologique
		Granularité	Atomique
		Visibilité	Boîte noire
Usage		Finalité	Découvrir
		Fonctionnalité	Rechercher, Classer
		Phase	Design time
Modèle	produit	Input utilisateur	-
		Requête	XQuery
		Descripteur	SAWSDL
		Input fournisseur	-

	processus	Nature d'ontologie	Domaine (input, output)
		Publication	-
		Formulation	Brut
		Appariement	Lexical, Sémantique
Implémentation		Technique de mesure	IR, Sémantique
		Usage d'ontologie	Interne (OWL-S)
		Architecture logicielle	Prototype expérimental

Comme l'approche précédente, l'approche SAWSDL-MX se centre sur un service atomique de type boîte noire selon une perspective technologique. L'originalité de cette approche est d'améliorer le processus de recherche de services sémantiques d'un annuaire par l'usage d'un appariement hybride (lexical et sémantique). Par conséquent, elle ne s'intéresse pas au problème de formulation de requêtes et publication de services. La requête attendue de l'utilisateur doit être conforme au langage XQuery. Les ontologies sont utilisées à l'intérieur de l'annuaire pour améliorer les performances de recherche.

2.3. L'approche SeCSE [Zachos et al., 2007]

SeCSE [Zachos et al., 2007] est une approche de découverte de services à partir des exigences exprimées en langage naturel. La principale fonction de l'approche est la recherche et le classement de services web à partir des requêtes composées et structurées en langage naturel. Cette approche utilise un annuaire étendu en définissant six facettes d'un service [Sawyer et al., 2005]. Chaque facette est décrite en utilisant une structure de données XML. La découverte de services basée sur les exigences utilise la facette « description de service ». La facette « qualité de service » est utilisée pour affiner la sélection une fois que les services sont découverts.

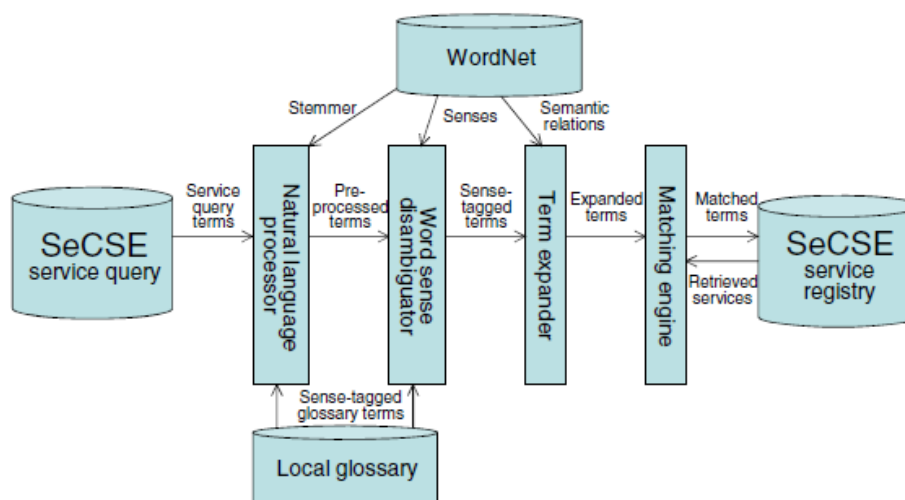


Figure 2.6. L'approche de recherche de services de SeCSE

La figure 2.6 montre que l'approche SeCSE est composée de 4 étapes. Dans la première étape, la requête de service permet de découper en phrases, puis en jetons, tagués et modifiés pour inclure les racines morphologiques de chaque terme (par exemple de la conduite à conduire, et du conducteur à conduire). Dans la seconde étape, l'algorithme applique des procédures pour lever l'ambiguïté en définissant à chaque terme son sens correct et l'insère pour chaque terme (par exemple la définition d'un pilote est le conducteur d'un véhicule plutôt qu'un pilote d'un périphérique informatique). Dans la troisième étape, l'algorithme augmente chaque terme avec d'autres termes qui ont un sens similaire, afin d'augmenter ainsi la probabilité d'un appariement avec une description de service (par exemple, conducteur est un terme synonyme du terme automobiliste qui est également inclus dans la requête). Dans la quatrième étape, l'algorithme fait un matching avec tous les termes élargis et-tagués de la requête. Ce qui aboutit à un ensemble semblable de termes qui décrivent chaque service candidat, exprimé en utilisant la facette de la description du service, dans le registre des services. SeCSE répond à la requête en 2 étapes: (i) XQuery recherche syntaxiquement à découvrir une première série de descriptions des services qui satisfont les contraintes de recherche; (ii) le modèle de recherche d'information de la forme traditionnelle d'espace vectoriel, renforcé par WordNet, affine et évalue les qualités de l'ensemble de services candidats. Cette approche permet en deux étapes de surmonter la limitation de recherche de XQuery basée sur le texte.

A propos de WordNet : Le lexique en ligne WordNet joue un rôle important pour les trois composants de l'algorithme. WordNet est une base de données lexicales inspirée par les théories psycholinguistiques de la mémoire lexicale de l'homme [Miller, 1993]. Il a deux caractéristiques importantes. Premièrement, il divise le lexique en quatre catégories: noms, verbes, adjectifs et adverbes. Les sens des mots, appelés sens, pour chaque catégorie sont organisés en ensembles de synonymes (synsets) qui représentent des concepts, et chaque synset est suivi par sa définition qui contient une phrase définissant, un commentaire optionnel et un ou plusieurs exemples. Deuxièmement, WordNet est structuré selon des relations sémantiques entre les significations de mots qui relient les concepts. Les relations entre les concepts tels que les relations hyperonyme et hyponyme sont représentées comme des pointeurs sémantiques entre les concepts liés [Miller, 1993]. Un hyperonyme est un terme générique utilisé pour désigner toute une catégorie de cas spécifiques. Par exemple, un véhicule désigne toutes les choses qui sont désignés séparément par les mots train, char, traîneau à chiens, avion, et automobile, et est donc un hyperonyme de chacun de ces mots. D'autre part, un hyponyme est un terme spécifique utilisé pour désigner un membre d'une classe, par exemple chauffeur, chauffeur de taxi et les automobilistes sont tous les hyponymes du conducteur. Une relation sémantique entre la signification des mots, comme un hyperonymie, relient les concepts.

En résumé, l'approche SeCSE est présentée dans le tableau 2.9.

Tableau 2.9. *Les caractéristiques de l'approche SeCSE [Zachos et al., 2007]*

Vue	Facette	Valeurs
Sujet	Entité	Technologique
	Granularité	Atomique
	Visibilité	Boîte noire

Usage		Finalité	Découvrir
		Fonctionnalité	Formuler, Rechercher, Classer
		Phase	Design time
Modèle	produit	Input utilisateur	<i>LN</i>
		Requête	<i>XQuery</i>
		Descripteur	<i>WSDL étendu</i>
		Input fournisseur	-
		Nature d'ontologie	Générique (WordNet)
	processus	Publication	-
		Formulation	Intelligent
		Appariement	Lexical
Implémentation		Technique de mesure	IR
		Usage d'ontologie	Externe
		Architecture logicielle	Prototype expérimental

Comme les deux approches précédentes, l'approche SeCSE se centre sur un service atomique de type boîte noire selon une perspective technologique. L'originalité de cette approche est d'améliorer le processus de la formulation par l'expansion de la requête en utilisant WordNet. Le processus de formulation est intelligent puisqu'il transforme la requête exprimée en langage naturel selon quatre étapes successives : l'analyse lexicale, la capture sémantique de sens de la phrase, l'expansion des termes contenus dans la phrase et la formulation de ces phrases en XQuery. Néanmoins, celui-ci est automatique (non interactive) et n'est pas guidé.

Comme les deux autres approches, SeCSE se focalise uniquement sur la découverte de services et ne s'intéresse pas au processus de publication. Contrairement à l'approche précédente, l'appariement proposé n'est que lexical.

2.4. L'approche de [Da Silva et al., 2009]

L'approche de [da Silva Santos et al., 2008] et [da Silva Santos et al., 2009] pour la recherche et l'invocation de services est basée sur la modélisation de but, et suppose que les acteurs impliqués (les applications clientes, les fournisseurs de services, la plate-forme) sont placés dans un domaine commun et connu. Cette exigence est nécessaire parce que la démarche repose sur la disponibilité des ontologies de domaines spécifiques. Pour chaque domaine, des buts valables pour les acteurs de ce domaine sont identifiés ainsi que les tâches nécessaires à leur accomplissement. La figure 2.7 illustre les principaux éléments de cette approche :

- *La plate-forme de services* support les activités des fournisseurs (la publication) et les activités des utilisateurs (la recherche de services, la composition, l'invocation et la surveillance).

- *Le méta-modèle de service basé sur les buts* définit les concepts indépendants du domaine comme : le service, les intervenants, l'organisation, le but et la tâche, et leurs relations. Ces concepts sont utilisés par la suite et sont spécialisés dans les ontologies de domaine et de tâches.
- *Les ontologies de domaine* définissent les concepts du domaine spécifique et leurs relations. L'ontologie de domaine est exploitée par la plate-forme de service lors de la recherche de la publication (externe à l'annuaire).
- *Les ontologies de tâches* sont associées aux ontologies de domaine, elles fournissent les descriptions des tâches du domaine. En particulier, les relations de dépendance et décomposition entre les tâches.
- *L'ontologie de services basée buts (GSO)* permet de répertorier les services significatifs du domaine en décrivant en autres, le but qu'il permet d'atteindre et les tâches qu'il réalise. Celle ci est exploitée par la plateforme pour proposer à l'utilisateur final une recherche à base de buts.

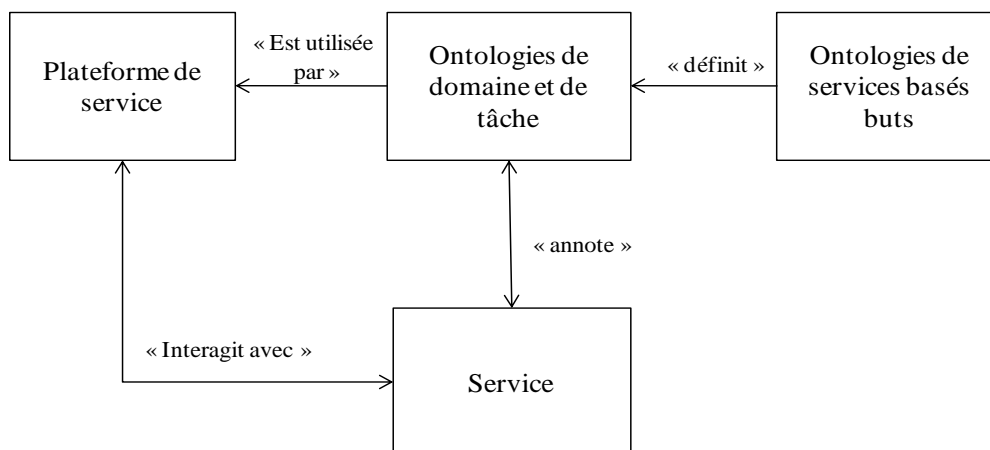


Figure 2.7. Les composants essentiels de l'approche et leurs relations

Les utilisateurs de service peuvent demander un service à la plate-forme qui répond à ce but, ou qui réalise une tâche. Dans le cas où un but est défini et soumis à la plate-forme, la plate-forme exécute la procédure suivante :

- Elle essaie de faire correspondre le but de l'utilisateur avec les buts définis dans l'ontologie de domaine ;
- Une fois qu'un but est trouvé, la plate-forme essaie de trouver des tâches définies dans l'ontologie de tâche qui peuvent atteindre ce but ;
- Avec la correspondance entre les services et les tâches établie à l'étape 2 ci-dessus, la plate-forme permet d'identifier un service disponible qui implémente la tâche qui répond au but demandé.

Dans le cas où une tâche est définie et soumise à la plate-forme, la plate-forme tente d'abord de trouver une tâche correspondante dans l'ontologie des tâches, puis les services qui implémentent le comportement de la tâche.

En résumé, l'approche est présentée dans le tableau 2.10.

Tableau 2.10. *Les caractéristiques de l'approche [Da Silva et al., 2009]*

Vue		Facette	Valeurs
Sujet		Entité	Technologique
		Granularité	Atomique
		Visibilité	Boite noire
Usage		Finalité	Découvrir, Composer, Invoquer
		Fonctionnalité	Formuler, Rechercher, Classer
		Phase	Design time
Modèle	produit	Input utilisateur	Tâche, But
		Requête	?
		Descripteur	WSMO
		Input fournisseur	-
		Nature d'ontologie	Générique (but), domaine (tâche)
	processus	Publication	-
		Formulation	Brut (tâche), Intelligent (but)
		Appariement	Sémantique
Implémentation		Technique de mesure	IR, Sémantique
		Usage d'ontologie	Externe (but), Interne (tâche),
		Architecture logicielle	Prototype expérimental

Comme les trois approches précédentes, l'approche Da Silva se centre sur un service atomique de type boite noire selon une perspective technologique. Da Silva propose une approche orientée services basée sur le concept de tâche. Une tâche est opérationnalisée par un ou plusieurs services. Son originalité est de proposer la recherche, la description, la publication et la composition de services dirigées par les tâches.

Le processus de recherche de service dirigé par les tâches utilise une ontologie des tâches en interne avec un appariement sémantique.

De plus, l'approche propose aux utilisateurs de rechercher des services à partir des buts. Dans ce cadre là, la plateforme propose un processus de formulation plus sophistiqué (intelligent) à base d'ontologie GSO (usage externe) pour traduire la requête initiale sous forme de buts en requête sous forme d'actions.

2.5. L'approche YASA [Chabeb et al., 2009]

Dans cette approche sémantique pour l'appariement des services web [Chabeb et al., 2009], les auteurs proposent un algorithme d'appariement basé sur les standards de services web, cet algorithme utilise les annotations sémantiques étendues définies par le langage YASA4WSDL [Chabeb et Tata, 2008]. Cette extension vient combler un manque au niveau de la spécification de la nature de l'annotation sémantique dans SAWSDL. La description YASA4WSDL est enrichie par des références portées sur des concepts techniques de services tels que *précondition*, *effet*, *opération*, etc. Ils utilisent deux types d'ontologies : (i) *ontologie technique* qui décrit la sémantique des concepts d'un service Web, et (ii) *ontologie de domaine* qui décrit les concepts d'un domaine métier particulier pour les services web.

Dans [Chabeb et al., 2009], ils définissent l'ontologie technique de descripteurs de services web qui intègre les concepts du méta-modèle WSDL[W3C, 2001], ceux de l'ontologie de OWL-S [Martin et al., 2004b] et WSMO [Lara et al., 2004]. L'intégration de ces ontologies est réalisée avec différentes techniques d'appariement de concepts. L'ontologie résultante offre une couverture sémantique large des concepts spécifiques aux services Web.

L'idée principale de YASA4WSDL est d'étendre SAWSDL afin d'améliorer l'expressivité de la description de service. Dans SAWSDL et pour chaque élément WSDL, on peut référencer plusieurs concepts d'une ontologie de domaine. L'annotation d'une opération par un concept d'un domaine métier est une annotation à gros grain, puisqu'elle ne distingue pas s'il s'agit d'une pré-condition, d'un effet ou d'un résultat de l'opération. Ainsi, YASA4WSDL ajoute dans le système d'annotation des descriptions de service Web SAWSDL un nouvel attribut nommé *serviceConcept*. Cet attribut permet l'annotation de plusieurs aspects d'un élément WSDL en utilisant des concepts techniques d'une ontologie de service, auxquels correspondent des concepts d'une ontologie de domaine référencés dans La facette « *modelReference* » de SAWSDL.

L'algorithme d'appariement se compose de trois variantes reposant sur trois différents degrés d'agrégations pour l'appariement sémantique. Cet algorithme a été implémenté en YASA-M [Chabeb et al., 2010], et il utilise le calcul de similarité pour l'appariement de services : terminologique, linguistique à base d'informations auxiliaires (WordNet), sémantique (Top-Down, Bottom-up et de voisinage) et structurel. Il utilise aussi trois types de filtres : structuraux, sémantiques et à base de seuil.

En résumé, l'approche YASA est présentée dans le tableau 2.11.

Tableau 2.11. Les caractéristiques de l'approche YASA [Chabeb et al., 2009]

Vue	Facette	Valeurs
Sujet	Entité	Technologique
	Granularité	Atomique
	Visibilité	Boite noire
Usage	Finalité	Découvrir
	Fonctionnalité	Rechercher, Classer
	Phase	Design time

Modèle	produit	Input utilisateur	?
		Requête	RDF
		Descripteur	SAWSDL étendu
		Input fournisseur	-
		Nature d'ontologie	Domaine, Technique
	processus	Publication	-
		Formulation	Brut
		Appariement	Lexical, Sémantique
Implémentation	Technique de mesure		IR, Sémantique
	Usage d'ontologie		Interne
	Architecture logicielle		Prototype expérimental

Comme les quatre approches précédentes, l'approche YASA se centre sur un service atomique de type boîte noire selon une perspective technologique. L'originalité de cette approche est de proposer une implémentation d'annuaire dont la sémantique est étendue en fonction d'un modèle (ontologie technique) tout en fournissant un algorithme hybride efficace d'appariement (lexical et sémantique). Par conséquent, elle ne s'intéresse pas aux problèmes de formulation de requêtes et publication de services. L'annotation sémantique intégrée dans le descripteur de service consiste en un double pointeur : le premier correspondant au concept sémantique et le deuxième correspondant à sa valeur dans l'ontologie de domaine.

2.6. L'approche SATIS [Mirbel et Crescenzo, 2009]

L'approche sémantique SATIS proposée par [Mirbel et Crescenzo, 2009] [Corby et *al.*, 2009a] repose sur un modèle pour capitaliser, réutiliser et partager des requêtes de recherche d'information et pour les organiser en un ensemble de démarches de recherche formalisées également réutilisables et partageables. Cette approche repose sur les techniques et modèles du Web sémantique, ainsi que sur la représentation intentionnelle d'un processus telle que proposé dans [Rolland et *al.*, 1999] [Rolland, 2007]. Elle tire parti des capacités d'inférence des ontologies capitalisant la connaissance d'un domaine d'application, et elle utilise le langage SPARQL afin de représenter les requêtes. Les auteurs proposent ainsi un modèle pour formaliser et raisonner sur les buts et les sous-buts. Enfin, cette proposition permet la réutilisation et le partage d'étapes (i.e. de sous-buts) entre les procédures de recherche. Précisément, les auteurs s'intéressent à la construction et l'exploitation d'une base de requêtes rendant opérationnelles les étapes d'un scénario de recherche d'information qu'ils appellent démarche de recherche d'information. Cette base de connaissances peut être vue comme une mémoire épisodique dans laquelle les démarches de recherche sont construites dynamiquement en fonction du contexte.

Les auteurs s'appuient sur les modèles et les langages du web sémantique pour proposer des moyens de raisonnement et d'explications des services web trouvés. Dans SATIS, trois

acteurs principaux sont déterminés : le concepteur de services Web, l'expert en modélisation de processus et l'expert du domaine métier.

- Le concepteur de services Web a pour mission de décrire les descriptions de services Web et de leur associer des spécifications de besoins intentionnels de haut niveau dans le but de promouvoir les services Web dont il a la charge.
- L'expert en modélisation de processus a pour mission d'enrichir une mémoire sémantique de la communauté construite avec les fragments de démarches dédiés à l'implémentation de besoins intentionnels de haut niveau.
- Enfin, l'expert du domaine métier (ou utilisateur final) recherche des descriptions de services Web qui lui permettent d'opérationnaliser un processus métier qui l'intéresse, une chaîne de traitement d'image dans le cas présenté par [Mirbel et Crescenzo, 2009]. Pour cela, il peut chercher parmi les besoins intentionnels déjà explicités dans la mémoire sémantique de la communauté, sinon il peut décider de créer par lui-même une nouvelle spécification. Cette étape de création consiste à spécifier des besoins intentionnels de haut niveau à l'aide des buts et des stratégies, et de les raffiner en buts et stratégies intermédiaires jusqu'à spécifier des buts suffisamment précis pour qu'ils soient associés à des patrons de spécification de services Web. Ensuite, l'étape de mise en œuvre de la démarche de recherche de services Web ainsi spécifiée consiste à opérationnaliser le processus métier (la chaîne de traitement d'image en l'occurrence), dont les besoins ont été élicités durant l'étape de création, à l'aide de services Web dont les annotations sont disponibles dans la mémoire collective de la communauté au moment de la recherche.

Cette approche s'appuie sur trois ontologies à la fois : (i) l'ontologie des processus intentionnels qui sont annotés à l'aide de concepts et des relations de cette ontologie permettant ainsi le partage et le raisonnement sur ces représentations intentionnelles, (ii) l'ontologie de domaine de neuroscientifique décrivant les images médicales et leurs traitements d'images associés, et (iii) la description sémantique de services web selon OWL-S [OWL-S, 2003].

Enfin, les requêtes de services web sont définies avec le langage de requête SPARQL [W3C, 2008]. Les annotations RDF, qui représentent les besoins intentionnels, et les requêtes en SPARQL, qui représentent les descriptions de services Web, sont ensuite rassemblées dans des règles. Celles-ci sont considérées comme des fragments de démarches de recherche de services Web réutilisables et forment la mémoire sémantique de la communauté.

L'étape de mise en œuvre est supportée par un moteur de chaînage arrière qui exploite les règles et les annotations de services Web. L'approche s'appuie sur le moteur sémantique CORESE [Corby et al., 2006] [Corby et al. 2009b] à la fois pour le mécanisme de chaînage arrière sur la base des règles de SATIS, et pour l'appariement des requêtes spécifiées dans les règles avec la base de connaissance des annotations OWL-S de services Web (base d'annotation des services de la communauté). Ainsi, un membre de la communauté qui cherche des services Web pour opérationnaliser une chaîne particulière de traitement d'images pourra tirer profit de toutes les règles et de toutes les annotations de services Web présentes dans la mémoire sémantique de la communauté au moment de sa recherche, en favorisant ainsi le partage, la réutilisation et la fertilisation croisée de savoir-faire, figure 2.8.

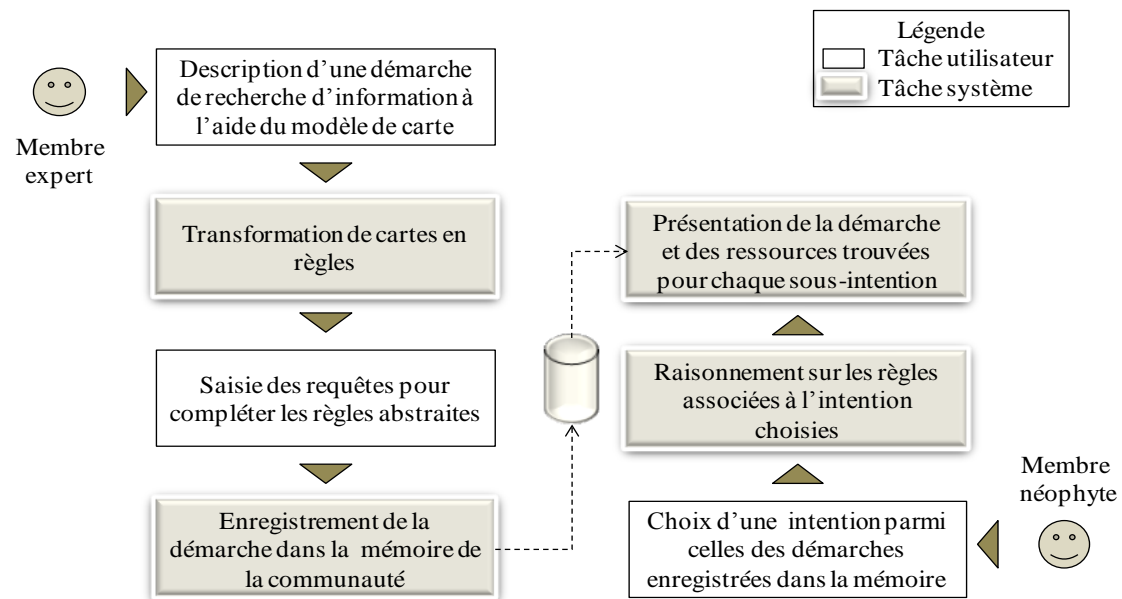


Figure 2.8. SATIS : Processus de définition et de mise en œuvre de démarches

En résumé, l'approche SATIS est présentée dans le tableau 2.12.

Tableau 2.12. Les caractéristiques de l'approche SATIS [Mirbel et Crescenzo, 2009]

Vue		Facette	Valeurs
Sujet		Entité	Métier
		Granularité	Atomique
		Visibilité	Boite noire
Usage		Finalité	Découvrir, Composer
		Fonctionnalité	Formuler, Rechercher
		Phase	Design time
Modèle	Produit	Input utilisateur	Modèle de but, Modèle de Carte
		Requête	SPARQL
		Descripteur	OWL-S
		Input fournisseur	-
		Nature d'ontologie	Domaine
	Processus	Publication	-
		Formulation	Guidé
		Appariement	Lexical

Implémentat ion	Technique de mesure	IR
	Usage d'ontologie	Interne (domaine), Externe (fragment)
	Architecture logicielle	Prototype expérimental

L'approche SATIS est la première approche de notre panel à s'intéresser à la perspective métier des services. Les services sont atomiques et de type boîte Noire, puisque c'est le modèle de Carte qui permet de composer selon le point de vue métier les services sélectionnés.

Il est à noter que l'utilisateur raisonne au niveau but et modèle de Carte pour formuler ses besoins mais cette vision but n'existe pas dans le descripteur OWL-S des services recherchés. C'est pour cette raison que orienté but n'est pas sélectionné comme valeur pour La facette entité.

Le processus de recherche est guidé et indépendant de l'annuaire de services sémantiques. SATIS prend en compte la découverte et la composition de services lors de la conception du système (design time). Le processus de publication et l'algorithme de l'appariement (lexical) sont laissés à la charge de l'annuaire de services. Par contre, les ontologies sont utilisées à l'intérieur et à l'extérieur de l'annuaire par le processus de recherche.

L'originalité de cette approche est (i) de proposer le modèle de Carte comme langage d'expression des besoins de l'utilisateur en termes de services et de composition de services, et (ii) de stocker ces compositions comme des fragments réutilisables dans la communauté de pratiques.

2.7. L'approche de [Driss et al., 2010]

Dans cette approche de modélisation, de recherche et de sélection de services web, les auteurs proposent une approche centrée exigence qui permet : (i) la modélisation des besoins des utilisateurs pour les applications basées services (SBA) en termes de besoins fonctionnels et non fonctionnels ; en utilisant le formalisme de la Carte et en spécifiant les services requis en utilisant le modèle intentionnel de service (MiS), (ii) la recherche des services pour interroger le moteur de recherche des services Web « Service-Finder » et en utilisant des mots-clés extraits des spécifications fournies par MiS, et (iii) la sélection automatique des services pertinents et de haute qualité en appliquant l'analyse formelle de concepts (AFC), figure 2.9.

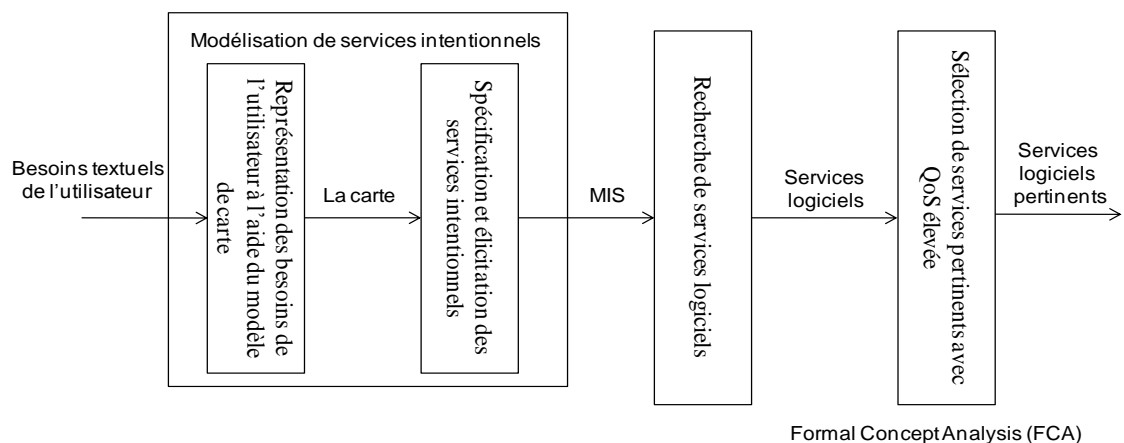


Figure 2.9. Les étapes de l'approche [Driss et al., 2010]

Dans la première étape, l'approche permet une modélisation orientée buts en utilisant le formalisme de la Carte [Rolland et *al.*, 2008]. Ensuite, MiS est utilisé afin de spécifier les services intentionnels présentés par les cartes.

Dans la deuxième étape, l'approche permet la recherche de services opérationnels en interrogeant le moteur de recherche *Service-Finder* en utilisant les mots-clés extraits de modèles MiS. Pour découvrir les services pertinents de manière efficace et de haute qualité, ils proposent le filtrage à deux niveaux. Au premier niveau, certaines propriétés de QoS sont examinées comme la « validité » (c.-à-d., on vérifie si l'URI du service est valable) et la « disponibilité » (ils vérifient si le service est opérationnel). Les services qui passent le premier niveau sont filtrés selon un appariement entre la spécification sémantique intentionnelle fournie par MiS et la spécification opérationnelle fournie par WSDL.

Dans la troisième étape, l'ensemble restant des services est classé dans une structure ordonnée appelée un treillis de concepts par l'application de l'analyse formelle de concepts (AFC) [Ganter et Wille, 1999]. AFC est un cadre formel qui permet de regrouper les individus qui partagent des propriétés communes et de les organiser en treillis de concepts. L'AFC permettra d'automatiser la tâche de sélection en fournissant une vue claire et organisée des services potentiels, ce qui permet aux utilisateurs de vérifier facilement si les services sont pertinents et de haute qualité.

En résumé, l'approche de Driss est présentée dans le tableau 2.13.

Tableau 2.13. *Les caractéristiques de l'approche [Driss et al., 2010]*

Vue		Facette	Valeurs
Sujet		Entité	Métier
		Granularité	Atomique
		Visibilité	Boîte noire
Usage		Finalité	Découvrir
		Fonctionnalité	Rechercher, Classer
		Phase	Design time
Modèle	produit	Input utilisateur	But
		Requête	Mots clés
		Descripteur	WSDL
		Input fournisseur	-
		Nature d'ontologie	Aucune
	processus	Publication	-
		Formulation	Guidé
		Appariement	Lexical

Implémentation	Technique de mesure	IR
	Usage d'ontologie	Aucun
	Architecture logicielle	Aucune

Cette approche considère également le service selon la perspective métier mais pas orienté but même cette approche utilise le modèle MiS, car le service est décrit selon une perspective opérationnelle et non intentionnelle (WSDL standard).

Son originalité réside dans l'exploitation du modèle de Carte et du modèle MiS lors de la formulation de requêtes (processus guidé). Elle délègue la recherche au composant « Service-Finder » en lui passant les éléments MiS comme une séquence de mots-clés sans sémantique.

Comme toutes les autres approches, elle ne s'intéresse pas au processus de publication et de description de service.

3. RECAPITULATIF DE L'EVALUATION

Le cadre de référence que nous proposons permet d'étudier tous les aspects pertinents d'une approche de publication et de recherche de services. Les quatre vues qui le composent correspondent à quatre perspectives selon lesquelles on peut étudier la publication et la recherche des services : le *quoi*, le *pourquoi*, le *comment* et le *moyen*. Dans chacune de ces vues sont définies des facettes qui précisent les caractéristiques des approches. Nous avons illustré le cadre en étudiant et en positionnant sept approches représentatives de l'état de l'art.

Le tableau 2.14, synthétise les résultats de cette analyse.

Tableau 2.14. Résumé du positionnement des approches

Vue		Facette	[Gomez et al., 2006]	[Klusch et al., 2006]	[Zachos et al., 2007]	[Da Silva et al. 2009]	[Chabeb et al., 2009]	[Mirbel et al., 2009]	[Driss et al., 2010]	PASIS 2011
Sujet		<i>Entité</i>	Technique	Technique	Technique	Technique	Technique	Métier	Métier	Orienté but, métier
		<i>Granularité</i>	Atomique	Atomique	Atomique	Atomique	Atomique	Atomique	Atomique	Atomique, agrégat
		<i>Visibilité</i>	B.N	B.N	B.N	B.N	B.N	B.N	B.N	B.N, B.B
Usage		<i>Finalité</i>	Découvrir	Découvrir	Découvrir	Découvrir, Invoquer, composer	Découvrir	Découvrir, Composer	Découvrir	Découvrir, Explorer
		<i>Fonctionnalité</i>	Formuler, Rechercher	Rechercher, Classer	Formuler, Rechercher, Classer	Formuler, Rechercher, Classer	Rechercher, Classer	Formuler, Rechercher	Formuler, Rechercher, Classer	Décrire, Publier, Rechercher, Formuler, Classer
		<i>Phase</i>	Design time	Design time	Design time	Design time, Run time	Design time	Design time	Design time	Design time
Modèles	Produit	<i>Input utilisateur</i>	LN (but)	-	LN	Tâche, but	-	Modèle de Carte, modèle de but	Modèle de Carte, MiS	Modèle but
		<i>Requête</i>	Mots clés	XQuery	XQuery		Triplet RDF	SPARQL	Mots clés	Mots clés typés
		<i>Descripteur</i>	WSML	SAWSDL	WSDL	WSMO	SAWSDL étendu	OWL-S	WSDL	SAWSDL étendu (MiS)
		<i>Input fournisseur</i>	-	-	-	-	-	-	-	MiS (orienté but)
		<i>Nature ontologie</i>	Domaine	Domaine (Input, Output)	Générique (WordNet)	Domaine (tâche), générale (but)	Domaine, technique	Domaine	Aucune	Technique, Domaine, Générique (verbe)
	Processus	<i>Publication</i>	-	-	-	-	-	-	-	Guidé intelligent
		<i>Formulation requête</i>	Intelligent	Brut	Intelligent	Intelligent (but), Brut (tâche)	Brut	Guidé	Guidé	Guidé intelligent
		<i>Appariement</i>	Lexical	Sémantique, lexical	Lexical	Sémantique	Sémantique, lexical	Lexical	Lexical	Lexical
Implémentation		<i>Technique de mesure</i>	IR	IR, Sémantique	IR	-	IR, Sémantique	IR	IR	IR
		<i>Usage ontologie</i>	Externe	Interne (OWL-S)	Externe	Interne (tâche), Externe (but)	Interne (technique, domaine)	Interne (dom), Externe (fragment)	Aucune	Interne, Externe
		<i>Architecture logicielle</i>	Prototype	Prototype	Prototype	Prototype	Prototype	Prototype	Aucune	Prototype

Dans la vue **sujet**, nous notons que cinq approches [Gomez et *al.*, 2006], [Klusch et *al.*, 2009], [Zachos et *al.*, 2007], [Da Silva Santos et *al.* 2009] et [Chabeb et *al.*, 2009] considèrent le service selon une perspective Technologique. Alors que les deux autres approches [Mirbel et Crescenzo, 2009] et [Driss et *al.*, 2010] sont orientés vers les buts de l'utilisateur final, et selon une vision Métier des services. Néanmoins, aucune des 7 approches n'a exploité l'orientation but dans la définition du service. On peut également noter que les services sont vus dans toutes les approches comme des éléments Atomiques de type Boite Noire. Ceci démontre que la structure de composition des services publiés et recherchés n'est pas exploitée lors du processus de publication ou de recherche.

En termes d'**usage**, la finalité de toutes les approches est essentiellement la découverte de services. Uniquement [Da Silva Santos et *al.*, 2009] s'intéresse à l'invocation de services, et seulement [Mirbel et Crescenzo, 2009] et [Da Silva Santos et *al.*, 2009] s'intéressent à la composition de services. Les deux approches [Klusch et *al.*, 2009] et [Chabeb et *al.*, 2009] centrent leurs efforts sur la recherche et le classement des services selon une requête, alors que les 5 approches [Zachos et *al.*, 2007], [Da Silva Santos et *al.* 2009] et [Mirbel et Crescenzo, 2009] considèrent la formulation comme une étape préliminaire à la recherche. Il est à noter qu'aucune approche ne traite plus spécifiquement la description, la publication et l'exploitation de services. Du point de vue phase, les approches sont toutes utilisées lors de la conception du système (Design time), seul [Da Silva Santos et *al.*, 2009] offre une fonctionnalité d'invocation qui est utile au Run time.

Dans la vue **modèle**, on distingue les approches qui proposent une étape de formulation car elles ont un « modèle d'input utilisateur » et un « modèle de requêtes » qui est différent. Le processus est dit Intelligent lors qu'elles exploitent une ontologie [Gomez et *al.*, 2006], [Zachos et *al.*, 2007] et [Da Silva Santos et *al.*, 2009], et Guidé lorsqu'elles supportent l'utilisateur durant ce processus [Mirbel et Crescenzo, 2009] et [Driss et *al.*, 2010]. La facette « Nature d'ontologie » permet de spécifier le type d'ontologie lors de la formulation. [Gomez et *al.*, 2006], [Da Silva Santos et *al.* 2009] et [Mirbel et Crescenzo, 2009] utilisent une ontologie de domaine, alors que [Zachos et *al.*, 2007] utilise WordNet pour capturer le sens des mots. Les deux approches [Klusch et *al.*, 2009] et [Chabeb et *al.*, 2010] ne proposent pas d'étape de formulation, elles ont un « input utilisateur » à '-' et un « processus de formulation » à Brut puisque celle-ci se résume à exprimer la requête. Par contre, ces deux approches se centrent sur la fonctionnalité de Rechercher en proposant un appariement hybride (Lexical, Sémantique) entre une requête (en XQuery pour [Klusch et *al.*, 2009] et RDF pour [Chabeb et *al.*, 2009]) et un descripteur de services (en SAWSDL pour [Klusch et *al.*, 2009] et en SAWSDL étendu pour [Chabeb et *al.*, 2009]). L'originalité de [Chabeb et *al.*, 2009] est d'exploiter en plus d'une ontologie de Domaine une ontologie Technique qui implémente l'extension du descripteur selon une sémantique spécifique. Pour les autres approches qui ont une étape de formulation, elles sont massivement sur un appariement Lexical sauf [Da Silva Santos et *al.* 2009] qui est encore sémantique car il exploite l'ontologie de tâche et les annotations sémantiques dans la fonction Rechercher. Il est à noter que ces approches ne traitent pas la publication ; c'est pour cette raison que la facette « input fournisseur » est à '-' et la facette « publication » au niveau processus est à '-'.

Dans la vue **implémentation**, l'appariement est basé sur des techniques de mesures. Par conséquent, l'appariement lexical est basé sur des mesures de type IR alors que l'appariement

hybride est basé sur des mesures IR combinés à des mesures de types sémantiques. Enfin, la facette « usage d'ontologie » permet de savoir si l'ontologie est utilisée dans l'annuaire durant la recherche (interne) ou lors de la formulation (externe). Nous avons des approches qui comportent l'usage des ontologies comme dans [Mirbel et Crescenzo, 2009] où l'ontologie de domaine est utilisée pour la recherche (interne), alors que l'ontologie de fragments est utilisée par la formulation (externe). De même [Da Silva Santos et *al.*, 2009] utilise l'ontologie de tâche à la recherche et l'ontologie de but durant la formulation.

4. CONCLUSION

Ce chapitre a présenté un état de l'art sur les approches de publication et de recherche de services. Nous avons proposé un cadre de référence pour analyser chacune d'elles. Les questions qui ont été posées et qui ont abouti au cadre de référence que nous avons présenté correspondent à une vision analytique des approches de publication et de recherche de service. L'élément central dans cette analyse est le lien entre d'une part l'univers sémantique orienté métier de l'utilisateur, et d'autre part, les informations qui sont stockés dans les descripteurs de service. Ce lien est concrétisé de différentes manières au moment de la publication d'un service et lors de la recherche de service. Il repose sur l'usage de modèles sophistiqués pour saisir la requête du client et raisonner sur celle-ci, ainsi que sur l'usage d'ontologies de plusieurs sortes (générique, technique ou de domaine) pour pouvoir donner du sens aux éléments d'information qui apparaissent autant dans la requête de l'utilisateur que dans le descripteur de service. Enfin, on constate que les approches laissent de côté les problématiques liées à la publication et se focalisent sur la découverte de services. On peut classer ces approches en 3 catégories :

- (1) Celles qui sont centrées sur un service technique [Klusch et *al.*, 2009] et [Chabeb et *al.*, 2010], elles ne considèrent la découverte que du point de vue de la recherche en améliorant celle-ci significativement sur les techniques d'application et sur la stratégie d'extension sémantique du descripteur.
- (2) Celles qui sont centrées sur un service technique [Gomez et *al.*, 2006], [Da Silva Santos et *al.*, 2009] et [Zachos et *al.*, 2007] qui proposent une étape de formulation préliminaire à la recherche pour traiter plus efficacement la découverte de services. Ces approches proposent des traitements intelligents de l'input utilisateur afin de formuler la requête nécessaire à l'algorithme de recherche. Le processus de formulation est intelligent mais pas guidé puisque l'utilisateur n'interagit pas.
- (3) Celles qui sont centrées sur un service plus métier [Mirbel et Crescenzo, 2009] et [Driss et *al.*, 2010] proposent un processus de formulation guidé par l'usage de modèles intermédiaires de haut niveau permettant graduellement d'exprimer la requête attendue par l'algorithme de recherche.

Pour la découverte de services, nous pouvons constater qu'aucune approche ne propose de fonction d'exploration de services à l'utilisateur. Cependant, cette fonctionnalité est importante pour découvrir des possibilités que l'utilisateur n'a pas envisagées. C'est pour cette raison que notre proposition *PAIS* supporte l'exploration (fonctionnalité = Explorer) et la structure de composition des services (Métier, Boîte Blanche) est exploitée dans ce cadre là.

Nous proposons à l'utilisateur en plus de l'exploration, les fonctionnalités de Formuler, Rechercher et Classer. Néanmoins, nous proposons une approche qui tient compte des caractéristiques des 3 catégories précitées. En effet, nous introduisons un processus de formulation qui est intelligent comme dans la catégorie 1 et qui guide l'utilisateur au niveau

métier comme dans la catégorie 2 et qui applique les principes d'extension sémantiques de [Chabeb et *al.*, 2010] (catégorie 3).

Enfin, cet état de l'art met bien en évidence que les approches ne prennent pas en compte le processus de publication de services. Néanmoins, la description de services n'est pas une tâche triviale pour le fournisseur surtout, s'il veut avoir une description de services la plus fidèle et la plus proche des besoins de l'utilisateur. C'est pour cette raison que notre proposition fournit un support méthodologique au fournisseur pour la description et la publication du service (fonctionnalité = publier, décrire et publication = guidé, intelligent)

L'approch *PASiS* (**P**ublishing **A**nd **S**earching of intentional **S**ervices) que nous proposons, regroupe les caractéristiques précitées. Le tableau 2.15 montre le positionnement d notre approche par rapport au cadre de référence :

Tableau 2.15. Les caractéristiques de l'approche *PASiS*

Vue		Facette	Valeurs
Sujet		Entité	Métier, Orienté but
		Granularité	Atomique, Agrégat
		Visibilité	Boite noire, Boite blanche
Usage		Finalité	Découvrir, Explorer
		Fonctionnalité	Décrire, Publier, Formuler, Rechercher, Classer
		Phase	Design time
Modèle	produit	Input utilisateur	Modèle de But
		Requête	Mots clés typés
		Descripteur	SAWSDL étendu
		Input fournisseur	MiS (orienté but)
		Nature d'ontologie	Générique (verbes), domaine, technique
	processus	Publication	Guidé, Intelligent
		Formulation	Guidé, Intelligent
		Appariement	Lexical
Implémentation		Technique de mesure	IR
		Usage d'ontologie	Interne, Externe (verbes)
		Architecture logicielle	Prototype expérimental

Les valeurs de chacune des facettes sont détaillées au fur et à mesure dans les différents chapitres de cette thèse.

CHAPITRE 3

L'APPROCHE PASiS

1. INTRODUCTION

Ce chapitre présente l'approche *PASiS* (**P**ublishing **A**nd **S**earching intentional **S**ervices) qui est une continuation des travaux antérieurs sur l'architecture iSOA et le modèle de service intentionnel MiS. Il s'agit de rendre opérationnelle cette proposition de recherche en proposant une plateforme iSOA avec des mécanismes de publication et de recherche de services intentionnels à partir d'un annuaire étendu. L'approche propose un guidage méthodologique pour (i) publier le descripteur d'un service intentionnel et le processus de publication d'un tel service, (ii) formuler les besoins de l'utilisateur sous forme de buts et (iii) rechercher, explorer et sélectionner les services intentionnels les plus proches des besoins de l'utilisateur.

La vision globale de l'approche et sa pertinence par rapport aux problèmes identifiés au chapitre 1 et rappelés dans la section 1 sont discutées à la section 2.

La section 3 permet de s'attarder sur les éléments clefs (ou caractéristiques) de l'approche qui sont : l'expression orientée métier, le processus de guidage méthodologique, l'utilisation des ontologies et l'usage du degré de similarité.

Les processus de publication et de recherche de services utilisent très largement les ontologies pour capturer, du côté fournisseur, la sémantique du descripteur de service intentionnel et du côté utilisateur, le besoin à satisfaire. Les trois ontologies utilisées sont décrites à la section 4 : *iSOnto* (ontologie technique de services intentionnels), *vOnto* (ontologie générique des verbes) et *pOnto* (ontologie de domaine des produits).

Ce chapitre est consacré également, à la spécification et à l'implémentation du descripteur intentionnel de service où l'annotation est utilisée comme une technique d'extension sémantique vers les trois ontologies : *iSOnto*, *vOnto* et *pOnto*. Dans ce travail, nous proposons d'étendre le langage SAWSDL pour décrire les services intentionnels. Cette extension vient combler un manque au niveau de la spécification de la nature de l'annotation sémantique dans SAWSDL. Pour cette raison, nous allons utiliser l'ontologie *iSOnto* pour expliciter non

seulement les concepts d'un domaine mais également la nature de ces concepts. La section 5 explique l'extension intentionnelle du descripteur de service.

Enfin, les processus de guidage de l'approche *PASIS* sont formalisés dans les chapitres 4 et 5 à l'aide du Modèle de Carte (MAP) qui fait l'objet de la section 6 de ce chapitre.

Dans la suite de ce mémoire, nous faisons une distinction entre les termes « Carte » et « Map ». En effet, une carte indique un produit, c'est une instance du modèle de Carte vue comme un modèle de produit ; tandis qu'un Map désigne un modèle de processus, il est généré en utilisant le modèle de Carte en tant que modèle de processus.

Un résumé de l'approche et de ses apports sont rappelés en conclusion de ce chapitre (section 7).

2. L'APPROCHE *PASIS*

L'objectif principal de l'approche *PASIS* est (i) de guider les fournisseurs dans la publication des services intentionnels (décrits à l'aide d'annotations sémantiques à base d'ontologies), (ii) de permettre aux utilisateurs finaux d'exprimer leurs besoins sous forme de buts et (iii) d'aider les utilisateurs à rechercher les services intentionnels disponibles qui satisfont leurs besoins métier.

Avant de présenter notre approche, nous allons rappeler la problématique de cette thèse. Ensuite, nous présentons une vue globale de l'approche *PASIS*.

2.1. Rappel de la problématique

L'absence de vision métier dans les approches à base de services, et la nécessité de guidage dans la publication et la recherche de services, nous ont amené à énoncer les objectifs suivants : (i) étendre l'annuaire avec un descripteur de service intentionnel annoté sémantiquement, (ii) prendre en compte la granularité de services intentionnels telles que définies dans le modèle de MiS, (iii) modéliser les processus de publication et de recherche d'i-services et (iv) construire un algorithme d'appariement pour mesurer le degré de similarité entre la requête et les services disponibles dans l'annuaire.

2.2. Vue globale de l'approche

L'approche *PASIS* concerne la publication et la recherche de services intentionnels. Les éléments de l'approche sont classés en 2 colonnes (voir figure 3.1), ceux qui contribuent à la publication sur la colonne de gauche et ceux qui contribuent à la recherche sur la colonne de droite. Néanmoins, deux éléments sont communs à la publication et la recherche : les ontologies et l'annuaire étendu.

Pour surmonter le problème d'écart entre le niveau métier et le niveau logique à la fois au niveau de la publication et au niveau de la recherche de services et éventuellement, éviter la modification de la structure de l'annuaire, une couche conceptuelle est ajoutée entre la couche métier et la couche logique. Ainsi, l'approche proposée ne modifie pas la structure de l'annuaire, ni les algorithmes utilisés pour la recherche. Les éléments de l'approche sont alors situés à trois niveaux : métier, conceptuel et logique.

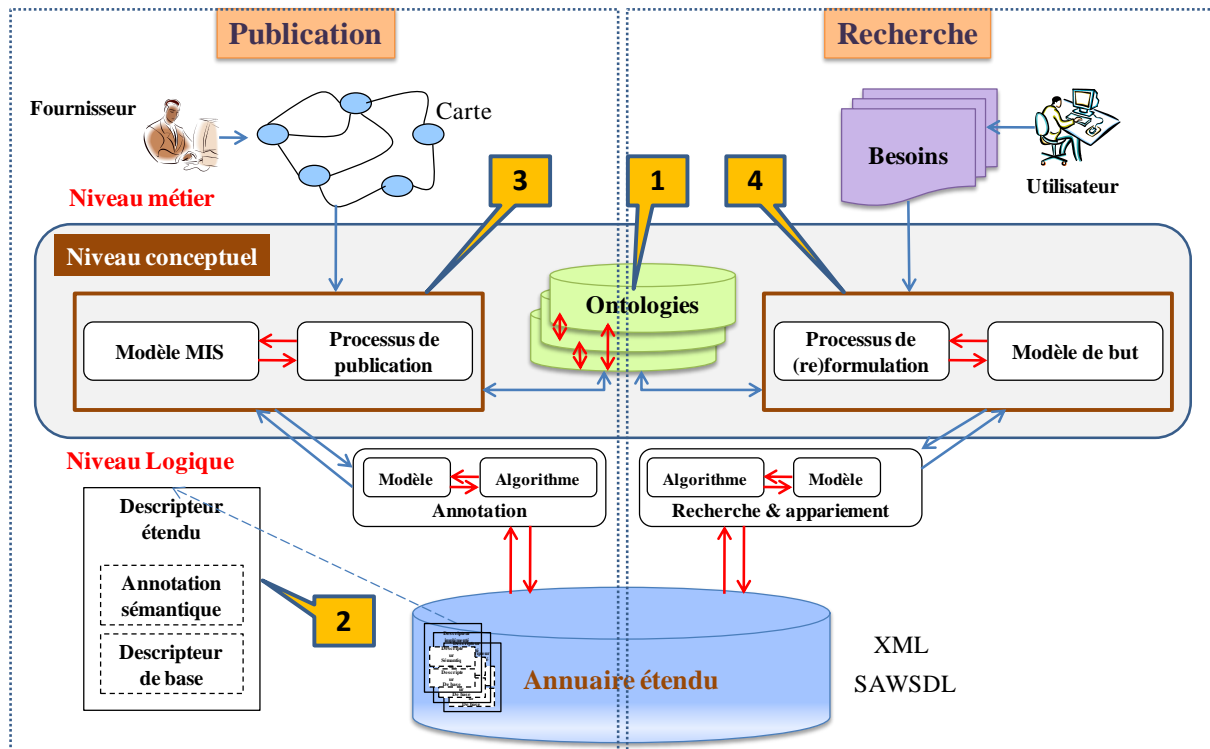


Figure 3.1. Vue globale de l'approche *PASIS*

L'essentiel de l'apport de l'approche *PASIS* se situe au niveau de la couche conceptuelle grâce aux modèles (MiS et but) et aux processus de guidage permettant de faire le lien entre le niveau métier et l'annuaire étendu (descripteur de service et algorithmes de recherche).

Une explication succincte du fonctionnement de l'approche est donnée dans la suite de cette section en expliquant (i) comment se déroulent la publication et la recherche de services, (ii) les ontologies et (iii) la répartition en trois niveaux.

- ✓ **La « publication de services »** consiste à enrichir l'annuaire *PASIS* par de nouveaux services intentionnels. Ces services sont extraits du cahier des charges des entreprises et ils sont exprimés en termes métier avec le formalisme de la Carte, puis annotés sémantiquement suivant un processus de guidage à l'aide d'ontologies [Aljoumaa et al., 2010]. Finalement, les annotations sont vérifiées syntaxiquement pour être conformes au modèle MiS et sémantiquement par rapport aux ontologies (*vOnto*, *pOnto*). L'annuaire étendu stocke alors, les intentions des services dans les descripteurs sous forme d'annotations sémantiques typées. Ces annotations sont utilisées au moment de la recherche de services. L'acteur principal de la publication est le *fournisseur* de services, il publie les services pour favoriser la réutilisation. Le modèle conceptuel MiS et le processus de guidage de la publication aident le fournisseur à décrire au plus près du métier la logique du service. De plus, la nature agrégée de ces services est supportée dans notre approche de publication puisque le modèle MiS et le descripteur intentionnel de service implémenté dans l'annuaire permettent de décrire des agrégations de services dirigées par les buts. Cette agrégation est exploitée à la publication et à la recherche de services.

- ✓ **La « Recherche de services »** permet à un utilisateur d'exprimer sa requête à l'aide d'un modèle de but. Dans *PASIS*, les problèmes d'un agent métier sont des problèmes de business exprimés au moyen de buts à atteindre ; par exemple, « payer une réservation d'hôtel », « effectuer un examen médical dans une clinique »... Ensuite, grâce au processus de guidage de recherche et d'interrogation, cette requête est reformulée à l'aide des ontologies pour avoir une liste de formulations. Ces formulations sont triées selon un degré de similarité et proposées à l'utilisateur. La formulation choisie est prise comme requête de recherche au niveau de l'annuaire. Les services trouvés sont transformés en modèle conceptuel et envoyés à l'utilisateur. Il s'agit de satisfaire le besoin exprimé en recherchant des candidats parmi les services intentionnels disponibles. La satisfaction d'un besoin peut nécessiter plusieurs services. A ce niveau, l'utilisateur obtient un service atomique ou un service agrégat qu'il peut explorer pour découvrir d'autres possibilités non prévues. L'acteur principal de la recherche est un *utilisateur métier* qui peut intervenir pour affiner la formulation ou explorer le résultat pour sélectionner les services qui satisfont son besoin. La recherche dans *PASIS* commence à partir d'un besoin utilisateur exprimé par un but qui est l'expression la plus proche du contexte métier de l'utilisateur. La nature interactive du processus de recherche avec l'utilisateur permet de proposer à la fois un mode de type recherche et un mode de type exploration où d'autres choix, auxquels il n'avait pas pensé, lui seront fournis. De plus, la partie intelligente de la recherche où l'on calcule d'autres formulations du besoin initial est effectuée par le processus de reformulation à l'extérieur de l'annuaire. La distance sémantique entre la reformulation proposée et le besoin initial est basée sur les liens sémantiques (généralisation et spécification) entre les concepts de l'ontologie de domaine (*pOnto*).

- ✓ **Les « Ontologies »** sont communes à la publication et à la recherche. Le rôle de ces ontologies est de disposer d'un vocabulaire commun pour spécifier les descripteurs des services intentionnels lors de la publication et de spécifier le but de l'utilisateur lors de la formulation de requête. Ces ontologies définissent une terminologie réutilisable et partageable par ceux qui fournissent les services et par ceux qui les recherchent. Les ontologies permettent aussi d'enrichir la sémantique de l'expression des requêtes. Ce niveau sémantique joue un rôle essentiel dans la recherche des services. Finalement, nous prenons en compte la structure du modèle de but (verbe, sujet, paramètres) dans les annotations sémantiques utilisées dans le descripteur d'un service. Pour cette raison, nous avons l'ontologie technique *iSOnto*, et une distinction importante est faite entre les noms propres (sujet, paramètres) et le verbe dans l'expression d'un but. L'usage de l'ontologie générique des verbes (*vOnto*) et l'exploitation des liens sémantiques entre les verbes et les noms propres (*pOnto*) permettent de raisonner sur la signification d'un but et guider plus intelligemment les processus de publication et de recherche de service. Ces trois ontologies sont utilisées lors de la publication pour valider la cohérence du but, et lors de la recherche pour trouver des formulations alternatives. Les ontologies, contrairement aux travaux sur les services sémantiques, ne sont pas utilisées lors de l'appariement mais par les processus de publication et de recherche au niveau conceptuel.

- ✓ **La « Répartition en trois niveaux »** (métier, conceptuel et logique) permet de faire le point entre le métier et la technique :
 - **Le niveau métier** : permet d'une part, à l'utilisateur d'exprimer ses besoins métier à la recherche de services intentionnels par une requête structurée sous forme de but. Cette

requête sera formalisée au niveau conceptuel. D'autre part, le fournisseur utilise le modèle de carte pour exprimer le processus métier et exploite le modèle MiS pour identifier les services candidats à la publication.

- **Le niveau conceptuel** : Ce niveau est le corps de notre travail. Son rôle consiste à guider l'utilisateur dans sa démarche de recherche et le fournisseur dans sa démarche de publication. Ce niveau contient aussi les ontologies utilisées dans les deux démarches. Ces démarches sont formalisées à l'aide du modèle de Carte. Pour cette raison, une première carte, appelée *carte de publication*, est définie. Cette carte guide la publication des services par la structuration de l'annotation sémantique des descriptions et la résolution des conflits. Enfin, une deuxième carte, appelée *carte de recherche*, est proposée. Cette carte guide la formulation de la requête représentant les besoins métier de l'utilisateur, elle la structure, la reformule et guide la recherche des services correspondants.
- **Le niveau physique ou logique** : correspond à l'annuaire étendu des services. Cet annuaire stocke les descripteurs selon plusieurs formes (standards ou intentionnels). La requête de l'utilisateur exprimée selon le modèle de but est transformée en requête logique conforme au modèle d'annotation du descripteur intentionnel. L'algorithme de recherche à base d'appariement compare le modèle logique de la requête avec le modèle logique du descripteur implémenté. Le résultat est envoyé au niveau conceptuel pour qu'il soit transformé en modèle compréhensible par l'utilisateur. A partir des buts de l'utilisateur, nous transformons la requête sous forme d'un descripteur de service demandé, et nous réalisons leur *appariement* avec les descripteurs des services disponibles. Pour cela nous utilisons des algorithmes existants basés sur les techniques d'analyse des similarités entre les éléments des requêtes de l'utilisateur et ceux des services intentionnels publiés dans l'annuaire. En fonction des différentes similarités détectées, une liste, appelée *liste des services candidats*, est construite. Cette liste représente des services intentionnels disponibles qui sont proches des besoins de l'utilisateur. Il est important de noter que pour une requête donnée, l'annuaire rend soit des services atomiques ou agrégats ayant une intention identique à la nouvelle formulation de la requête initiale.

Les principales caractéristiques de l'approche seront précisées à la section suivante.

3. CARACTERISTIQUES DE L'APPROCHE

L'approche *PASIS* est basée sur quatre caractéristiques : (i) la représentation unifiée des buts pour une expression orientée métier, (ii) le guidage dans le processus de publication et dans le processus de reformulation de requêtes, (iii) l'utilisation d'ontologies, et (iv) l'usage de degré de similarité. Dans cette section, nous présentons en détail ces caractéristiques et nous montrons comment elles permettent de satisfaire les objectifs définis à la section 2.1.

3.1. Expression orientée métier

Les buts constituent un élément essentiel dans notre approche. Nous utilisons une représentation des buts commune à la phase de publication et à celle de formulation de requête.

L'approche présente un but simple sous forme lexicale. Nous nous appuyons, dans ce cadre, sur la formulation des buts du modèle MiS [Kaabi, 2007], [Rolland *et al.*, 2008] qui repose sur une approche linguistique initialement développée par [Prat, 1997], [Prat, 1999]. Cette approche inspirée de la grammaire des cas de Fillmore [Fillmore, 1968] et des extensions de

[Dik, 1989] se fonde sur le fait que la sémantique d'un but est capturée par un verbe et des paramètres qui correspondent à des cas ou rôles associés au verbe. Ce formalisme permet de représenter des objectifs des utilisateurs et des services. Suivant cette approche, un but est donc formulé par :

But : <Verbe> <Cible> [<Paramètre>]*

Les deux premiers éléments de ce modèle sont obligatoires alors que les autres paramètres sont optionnels et sont utiles pour préciser l'intention et exprimer des informations supplémentaires telles que : la source, le bénéficiaire, la direction, le temps, etc. (Figure 3.2).

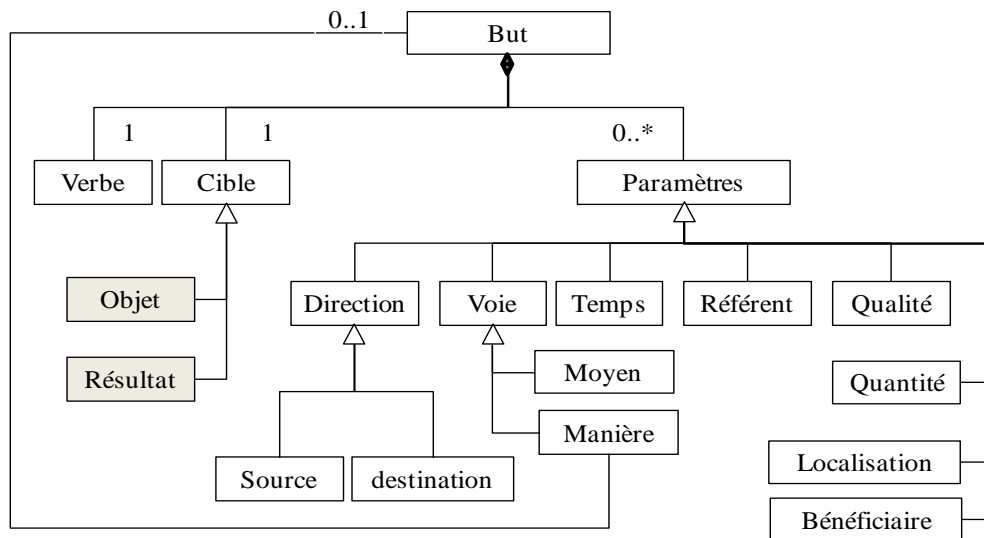


Figure 3.2. Le modèle de but

Dans l'exemple suivant : (*Effectuer*)_{verbe} (*une réservation*)_{résultat}, la réservation est l'objectif à produire par le verbe. Le modèle de but distingue entre deux types de cible : l'*Objet* et le *Résultat*. La réservation dans notre exemple est un résultat car elle est créée par les actions qui permettent de réaliser le but. Par contre, dans l'exemple (*Confirmer*)_{verbe} (*une réservation de chambre*)_{objet}, la cible est un objet puisque la réservation de chambre existe de manière préalable à la satisfaction du but.

Les paramètres du but sont les suivants :

La *Direction* : Il y a deux types de direction, la *Source* et la *Destination*.

La *Source* identifie l'endroit initial des objets à communiquer. Exemple : (*Etablir*)_{verbe} (le plan industriel et commercial)_{objet} (à partir des grandes orientations prévisionnelles)_{source}

La *Destination* identifie l'endroit final des objets à communiquer. Exemple : (*Stocker*)_{verbe} (*une réservation*)_{objet} (dans la base de données)_{destination}

Le *Bénéficiaire* est la personne, ou groupe de personnes, en faveur de qui le but doit être réalisé. Exemple : (*Réduire*)_{verbe} (les prix)_{objet} (pour les gros clients)_{bénéficiaire}

Le *Réfèrent* est l'entité par rapport à laquelle une action est effectuée ou un état est atteint ou maintenu. Exemple : (*Ajuster*)_{verbe} (les prix)_{objet} (au taux d'inflation)_{réfèrent}

La *qualité* définit une propriété qui doit être atteinte ou préservée. Exemple : (Rester) *Verbe* (plus fiable) *Qualité* (que les concurrents) *Référence*

La *Localisation* situe le but dans l'espace. Ce cas n'implique aucun mouvement dans le même endroit. Exemple : (Définir) *Verbe* (le plan de production prévisionnel) *Référence* (dans une unité de production) *Localisation*

Le *Temps* situe le but dans le temps. Exemple : (Expédier) *Verbe* (la commande) *Objet* (dans 8 jours) *Temps*

La *Quantité* indique la quantité. Exemple : (Réduire) *Verbe* (les prix) *Objet* (de 3%) *Quantité*

La Figure 3.2 montre enfin que seule la *Manière* est un paramètre complexe puisqu'une manière peut être elle-même un but. Par exemple, Améliorer (les services offerts aux clients d'une agence de voyages) *cible* (en mettant en place un système de réservation sur Internet) *manière* est un but dont la manière « *en mettant en place un système électronique de réservation* » est elle-même un but : (Mettre en place) *verbe* un (système de réservation) *résultat* (sur Internet) *moyen*.

Le formalisme de but sera utilisé d'un côté, pour exprimer les requêtes de l'utilisateur sous forme de buts métier, et de l'autre côté, le but rattaché au service intentionnel publié dans l'annuaire. Nous pensons que cette expression homogène permet d'éviter la discordance conceptuelle qui est au cœur de tous les problèmes de recherche des services.

Rappelons que la discordance conceptuelle vient du fait que les attentes et besoins de l'utilisateur sont naturellement exprimés dans une forme et à un niveau d'expression différents de ceux qui sont utilisés par les fournisseurs pour décrire les fonctionnalités des services.

3.2. Le guidage dans les processus

L'approche *PASIS* propose le guidage dans les deux principaux processus. Le premier guide le fournisseur lors de la publication. Le second guide l'utilisateur dans le processus de formulation de requêtes et de recherche de services.

Nous avons formalisé les deux processus guidés à l'aide du modèle de Carte. L'intérêt du modèle de Carte comme méta-modèle de processus est qu'il permet de représenter plusieurs démarches méthodologiques dans un même modèle de processus. Une carte, en tant que modèle de processus, est ainsi un modèle multi-démarches. Nous avons assorti chacun des processus d'un ensemble de directives méthodologiques. Ces directives assistent le fournisseur/utilisateur dans ses choix et dans le déroulement du processus choisi. Elles constituent un ensemble cohérent d'indications sur la façon de procéder pour réaliser un objectif ou exécuter une activité. Ces directives sont associées aux cartes des processus de publication et de recherche. Il existe trois types de directives :

- (1) *Directive de Réalisation d'Intention* (DRI) pour définir comment une intention peut être atteinte par une stratégie donnée et le mécanisme d'exécution de la tâche opérationnalisant cette intention. Une DRI est associée à une section.
- (2) *Directive de Sélection de Stratégie* (DSS) pour choisir une stratégie parmi un ensemble de stratégies données. Une DSS est associée à un couple d'intentions, l'une jouant le rôle de la source et l'autre celui de la cible.
- (3) *Directive de Sélection d'Intention* (DSI) pour découvrir toutes les intentions succédant une intention donnée. Une DSI est associée à une intention.

Le modèle de Carte et les types de directives sont détaillés dans la section 6.

3.2.1. Processus de publication

Ce processus guidé garantit que le descripteur soit proche en termes sémantiques de la signification et de la logique du service métier. D'autre part, il prend en charge la traduction de cette expression de haut niveau vers le niveau technique du descripteur manipulé par l'annuaire étendu.

Ce processus s'organise autour d'une activité principale qui est l'annotation de service. Cette activité consiste à ajouter les annotations sémantiques aux descripteurs de services. Les différentes stratégies représentées dans la carte de ce processus permettent de guider l'utilisateur dans sa démarche. Comme on le verra au chapitre 4, ce processus garantit la complétude de la description d'un service publié, et il prend en compte au niveau du descripteur, les agrégations de services intentionnels telles que définies dans le modèle MiS.

3.2.2. Processus de recherche

Ce processus guidé, d'une part, aide l'utilisateur à tirer profit de toutes les possibilités qu'offre le modèle de requête, et d'autre part, l'accompagne dans sa recherche et l'aide à trouver ce qu'il souhaite. Il s'organise autour de deux activités :

1. La formulation d'une requête
2. La proposition de services et l'exploration des services agrégats

L'activité (1) consiste à structurer la requête, à partir des éléments de base, selon le formalisme de but, à reformuler la requête, à trier les reformulations selon un algorithme d'appariement et les différentes stratégies pour les atteindre. Cette activité permet de construire une liste de formulations de la requête. Cette activité permet aussi de raffiner la requête pour filtrer les résultats.

L'activité (2) consiste à proposer les services en exploitant la structure des services intentionnels agrégés, l'utilisateur peut parcourir l'arborescence de services composites et à variation et explorer ainsi l'espace des services intentionnels disponibles. Cette activité permet aussi de sélectionner les services appropriés.

Nous avons formalisé le processus de recherche (chapitre 5) à l'aide du modèle de la Carte et d'un ensemble de directives méthodologiques plus détaillées permettant de guider l'utilisateur dans la recherche des services intentionnels.

3.3. Utilisation d'ontologies

Les ontologies définissent une terminologie pour partager un vocabulaire commun d'un domaine. Leur utilisation permet de représenter formellement les connaissances utilisées dans la description des services. Ces connaissances peuvent ensuite être exploitées pour affiner l'identification des services qui répondent au besoin de l'utilisateur, ce qui rend les résultats de la recherche de services plus pertinents.

Vu la structure du but, nous distinguons dans l'expression d'un but entre le verbe et les noms propres représentant la cible et les paramètres. Pour cette raison, nous utilisons une ontologie de domaine pour les noms propres et une ontologie générique des verbes. Les liens sémantiques entre les verbes et les noms propres permettent de raisonner ainsi sur la signification d'un but dans les processus de publication et de recherche de service. Une

troisième ontologie, cette dernière s'appelle une ontologie technique, est utilisée pour prendre en compte la structure du modèle de but (sujet, verbe, paramètres) qui est sous-jacente à MiS. Ainsi, la spécification des services est basée sur trois ontologies différentes qui donnent trois types de connaissances différentes : une ontologie de services intentionnels (*iSOnto*), une ontologie des verbes (*vOnto*) et une ontologie des produits (*pOnto*). Mais, les trois ontologies sont classées en deux types :

- **Les ontologies indépendantes d'un domaine (de haut niveau).** Ces ontologies décrivent les concepts généraux utilisés dans le modèle intentionnel de services. Elles sont indépendantes d'un domaine d'application comme l'ontologie technique de services intentionnels [Aljoumaa et al., 2011b] et l'ontologie des verbes [Urrego, 2005].
- **Les ontologies relatives aux domaines d'applications.** Elles définissent un vocabulaire qui décrit un domaine d'application particulier comme l'ontologie des produits [Aljoumaa et al., 2011a]. On suppose que les ontologies du domaine d'application sont disponibles et qu'elles sont réutilisables ; nous pouvons les utiliser en les important directement. Dans ce travail, et pour les besoins d'illustrations de l'approche *PASIS*, nous utilisons une ontologie du domaine médical (c.f. chapitre 6).

Ces trois ontologies fournissent un ensemble étendu de concepts pour annoter les services lors de la publication et pour exprimer les besoins des utilisateurs lors de la recherche. Elles assurent ainsi le partage de connaissances entre les fournisseurs de services intentionnels et ceux qui vont les chercher et les utiliser. La section 4 de ce chapitre présente en détail les ontologies utilisées dans l'approche *PASIS*.

3.4. Usage du degré de similarité

De nombreuses approches ont été proposées dans différents domaines pour mesurer les similarités : en ingénierie des système d'information dans le cadre de la réutilisation de composants, en génie logiciel pour assurer la traçabilité de code logiciel, dans le cadre d'analyse documentaire pour indexer et retrouver des documents, ou même pour l'analyse de schémas de bases de données hétérogènes [Bianco et al., 1999] [Castano et al., 1992], [Castano et De Antonellis, 1993] [Jilani et al., 1997] [Natt Och Dag et al., 2001] [Bouchachia et Mittermeir, 2001] [Salton et McGill, 1983] [Zoukar, 2005]. Ces différentes approches proposent des métriques exprimées comme des formules plus ou moins adaptées à différentes situations.

Dans l'approche, l'appariement consiste à faire correspondre les éléments constituant la requête (sous forme de but) et ceux des buts des services publiés dans l'annuaire, cet appariement se base sur l'égalité stricte entre les deux formulations de but. Par exemple, « confirmer la réservation » comme requête logique ne pourra pas avoir comme services candidats « confirmer la réservation par téléphone ». Comme l'appariement s'effectue au niveau logique (c.-à-d. au niveau de l'annuaire étendu), c'est en amont (au niveau conceptuel) que les ontologies sont utilisées pour élargir le champ sémantique de la requête et proposer des reformulations de celle-ci à l'utilisateur. Pour évaluer et trier les différentes reformulations proposées, il est nécessaire de calculer le degré de similarité entre une reformulation et le besoin initial.

Le degré de similarité est une fonction de calcul de distance basée sur les liens sémantiques entre concepts (généralisation, spécialisation) dans une ontologie. En d'autres termes, c'est la distance sémantique entre deux concepts dans la même ontologie. En effet, la distance sémantique de deux concepts peut être mesurée selon le type de liens entre ces deux concepts

dans une ontologie. Dans *PASIS*, cette mesure de similarité est appliquée sur les concepts de l'ontologie de domaine.

4. ONTOLOGIES DANS *PASIS*

Notre approche est basée sur l'utilisation de trois ontologies. La première est l'ontologie de services intentionnels qui contient les concepts définissant les éléments du service intentionnel (service, service atomique, agrégat, but, verbe, objet...) [Aljoumaa et al., 2011c]. La deuxième est l'ontologie de verbes classée par [Urrego, 2005]. La troisième, appelée Ontologie des produits, contient la sémantique des termes dans un domaine métier.

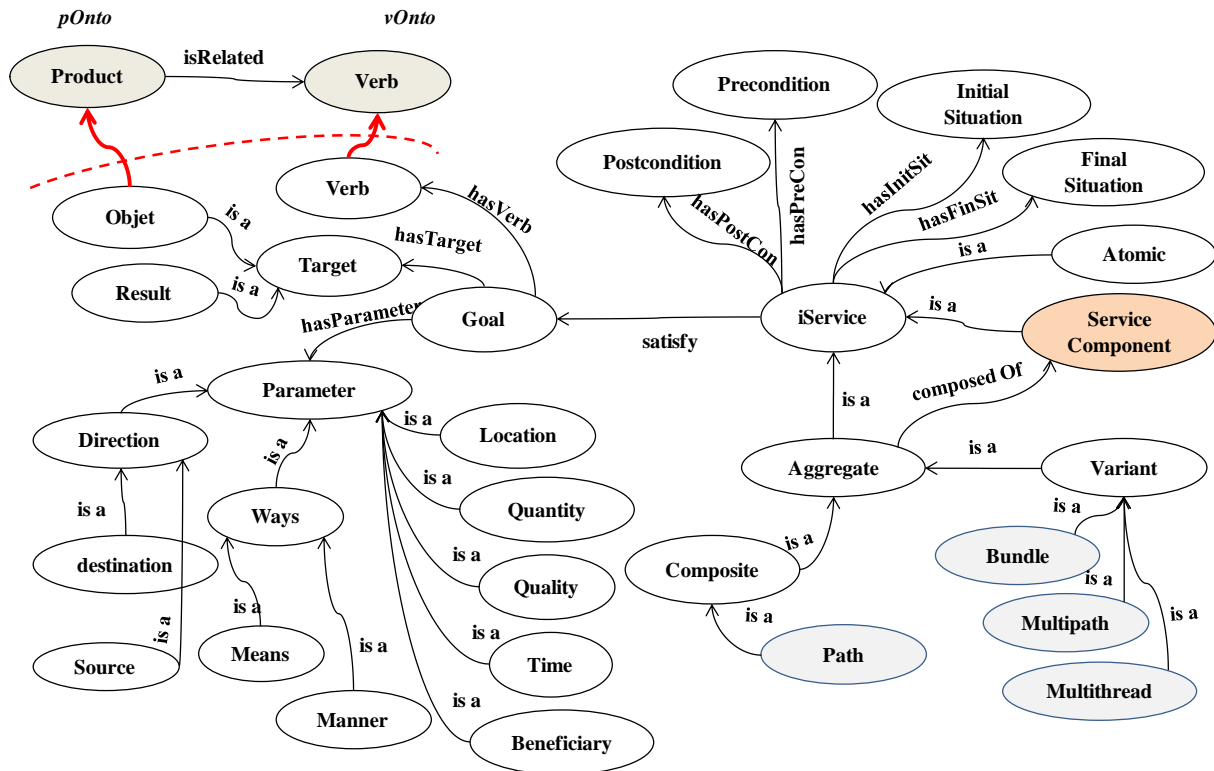
4.1. L'ontologie de services intentionnels (*iSOnto*)

L'ontologie *iSOnto* est représentée en termes de concepts et de liens entre eux. Les concepts sont des classes et les liens sont les propriétés d'objets. Les liens entre les concepts de l'ontologie sont des relations entre classes. Ces relations sont d'héritage, de composition ou d'affinement. Ainsi chaque concept de l'ontologie caractérise une classe et possède des relations avec les autres classes.

Nous avons construit l'ontologie de services intentionnels *iSOnto* pour annoter les descripteurs de services. Cette ontologie rassemble les concepts et les relations du modèle MiS et ceux qui participent à la définition de but. C'est une tentative de parvenir à un terrain d'entente entre les besoins des utilisateurs de haute niveau et le bas niveau de services logiciels, et de fournir la base d'interopérabilité et l'outil de mise en œuvre. En outre, l'ontologie *iSOnto* aidera également à construire une plateforme de recherche unifiée. Les utilisateurs recherchent les services qui satisfont leurs besoins exprimés selon les concepts de l'ontologie. Puis, notre plateforme est utilisée pour traduire la requête initiale en une requête spécifique conforme aux descripteurs de services intentionnels.

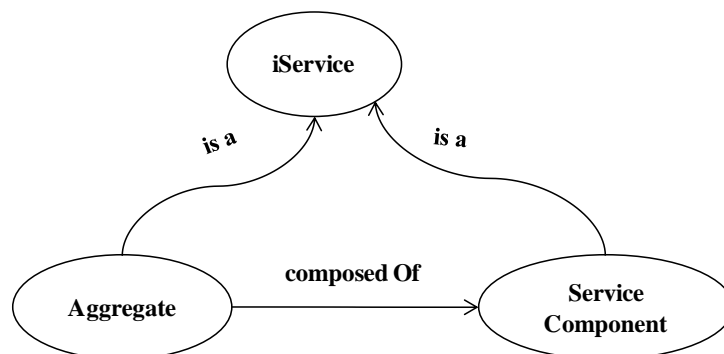
Cette section présente les concepts de l'ontologie et leurs relations. *iSOnto* a été construite sur le modèle de but et le modèle MiS (le fait que le service intentionnel est conçu comme un ensemble de services atomiques et de services agrégats).

On retrouve comme concepts principaux de cette ontologie *Service* et *Goal*. Un but est spécifié à l'aide d'un verbe, d'un objet et de paramètres (classes « *Verb* », « *Target* » et « *Parameter* » et des propriétés « *hasVerb* », « *hasTarget* » et « *hasParameter* »). Les sous-classes de la classe « *Parameter* » sont instanciées en différentes classes. La classe « *iService* » est spécialisée en trois-classes : « *Atomic* », « *Aggregate* » et « *ServiceComponent* » qui représente les composants d'un service agrégat, la figure 3.3.

Figure 3.3. L'ontologie de services intentionnels *iSonto*

Tout d'abord, nous avons ajouté la relation « *satisfy* » entre les deux modèles montrant que les besoins d'un utilisateur exprimés en tant que but sont satisfaits par les services intentionnels. Cela définit une correspondance parfaite entre le concept « *goal* » du modèle de but avec le concept « *goal* » de l'ontologie, et définit une parfaite adéquation entre la notion de « *service intentionnel* » du modèle MiS avec la notion de « *iService* » de l'ontologie.

Deuxièmement, nous avons ajouté une modification à la composition de services en *iSonto* concernant les composantes du service agrégat (composite ou à variation). Dans la figure 3.4, nous montrons le nouveau concept appelé « *serviceComponent* », sa relation « *isA* » avec le concept « *iService* », et sa relation « *composedOf* » avec le concept « *Aggregate* ».

Figure 3.4. Les nouveaux concepts dans *iSonto*

Troisièmement, la relation entre le « *verb* » d'*iSonto* et le « *verb* » d'ontologie des verbes *vOnto* montre comment cette ontologie est reliée avec *vOnto* et la relation entre le « *Cible* » d'*iSonto* et le « *Product* » d'ontologie des produits *pOnto* montre comment cette ontologie est reliée avec *pOnto*. La relation « *relatedTo* » entre le concept de « *verb* » de l'ontologie des verbes « *vOnto* » et le concept de « *Cible* » de l'ontologie des produits « *pOnto* » aide à l'enrichissement de la fonction sémantique et à la formulation de la requête de l'utilisateur [Aljoumaa et al., 2011a].

Enfin, la partie service est composée de la situation initiale, situation finale. Ces deux concepts sont respectivement appariés dans notre ontologie à la « *initial situation* » et « *final situation* ».

Pour résumer, chaque concept d'*iSonto* correspond dans l'ensemble des concepts émis du modèle de buts et du modèle de service intentionnel. L'ontologie représente un terrain sémantique commun utile pour réduire l'écart conceptuel entre le fournisseur et les besoins des utilisateurs qui sont exprimés en termes de métier et contribuera à définir le descripteur de services intentionnels et à formuler les requêtes des utilisateurs.

4.2. L'ontologie des verbes *vOnto*

L'ontologie de verbes utilisée dans notre approche représente des concepts sémantiques et syntaxiques relatifs aux verbes [Urrego, 2005].

L'ontologie est représentée en termes de concepts et de liens entre eux. Les concepts sont des *classes de sens de verbe* caractérisés par un *type* et par une *catégorie*. Ainsi chaque concept de l'ontologie qui définit un *type* et une *catégorie* de sens de verbe possède une *signature* différente. Chaque *sens de verbe* dans une classe caractérise une *classe de buts*. Les *liens* entre les concepts de l'ontologie sont des relations entre classes de *sens de verbe* et entre *classes de buts*. Ces relations sont d'héritage, de composition, d'affinement, de caractérisation ou des relations spécifiques d'un domaine.

Un verbe peut avoir plusieurs *sens* qui agissent comme s'il était un verbe différent. En conséquence, dans la suite, nous utilisons le terme de *verbe* en sous-entendant que cela indique un *sens* particulier de ce verbe. Un *sens* de verbe est une instance de la classe de *sens* de verbe qui est caractérisée par une *signature*.

Nous utilisons le concept de *but* pour représenter les phrases qui énoncent les requêtes des utilisateurs. Un but est composé par un *sens de verbe* et par un ensemble de *paramètres*. Ces *paramètres* déterminés par le verbe représentent, selon le modèle de but, les composants d'une phrase et aident à compléter le sens de cette phrase. La *cible* représente l'objet sur lequel agit le verbe. Le *moyen* et la *méthode* sont respectivement les instruments pour accomplir les actions du verbe et la façon dont celles-ci peuvent être réalisées. Le changement de la cible, due au verbe, est identifié par les paramètres *situations*.

4.2.1. Types de verbes

Les concepts supérieurs de l'ontologie définissent les quatre *types* de classes des sens de verbe *abstraites* c'est-à-dire des classes sans instances. Ces classes constituent les niveaux de *définition* de l'ontologie (figure 3.5).

Le *type* caractérise les paramètres du sens de verbe notamment celle du paramètre *cible*. Les Quatre types de verbes sont :

- *Causatif physique* indique une intervention *directe* et *physique* sur le paramètre *cible* : *accomplir (to effect)*, *accomplir* [la réservation d'hôtel] _{Cible}
- *Causatif intellectuel* indique une intervention *directe* et *intellectuelle* sur la *cible* : *afficher (to display)*, *afficher* [l'état de la réservation] _{Cible}
- *Opérationnel physique* indique une intervention *indirecte* et *physique* sur les relations avec la *cible* : *s'occuper (to deal with)*, *s'occuper* [des comptes] _{Cible}
- *Opérationnel intellectuel* indique une intervention *indirecte* et *intellectuelle* sur les relations avec la *cible* : *raisonner (to reason)*, *raisonner* [sur les choix] _{Cible}

4.2.2. Catégories de verbes

A ce niveau les classes de sens de verbe sont des classes concrètes contenant les instances qui énoncent les buts. Le concept *catégorie* caractérise le *sens d'un verbe* d'après les paramètres que celui-ci exprime. Cette caractérisation implique la caractérisation des paramètres en termes de paramètres *facultatifs* ou *interdits*, particulièrement le paramètre *situation*. Les catégories sont :

- *Catégories des verbes de service*. Un verbe de service énonce un développement par étapes. Les verbes de cette catégorie considèrent le changement de *situation*. Autrement dit, les verbes de service acceptent les paramètres *situation initiale* ou *situation finale* ou les deux : *fournir (Supplies)* [des services de tourisme] _{Target} [de la promotion des services] _{initial situation} [jusqu'à leur accomplissement] _{final situation} [au moyen d'un site web] _{Moyen}.
- *Catégories des verbes de processus*. Un verbe de processus exprime un développement par étapes au travers desquelles le paramètre *cible* change de *situation* : *traiter (process)* [le service d'hôtel] _{Cible} [du début] _{initial situation} [jusqu'à la fin] _{final situation} [au moyen du module de services du site web] _{moyen}.
- *Catégories des verbes d'activité*. Un verbe d'activité exprime un ensemble d'*actions* se réalisant en une seule étape. Il ne considère pas le changement de *situations* de leur *cible*.
- *Catégories des verbes d'action*. Un verbe d'action/état exprime un fait visant à produire un effet donné. Ils donnent le caractère *instantané*. Il ne considère pas le changement de *situations* de leur *cible*.

4.2.3. Signature de verbes

La *signature* caractérise une *classe de sens de verbe* selon les attributs *type* et *catégorie*. De cette manière, les quatre types et les quatre catégories de verbes définissent seize *classes de sens de verbe* fondamentales, chacune identifiée par une signature différente, la figure 3.5.

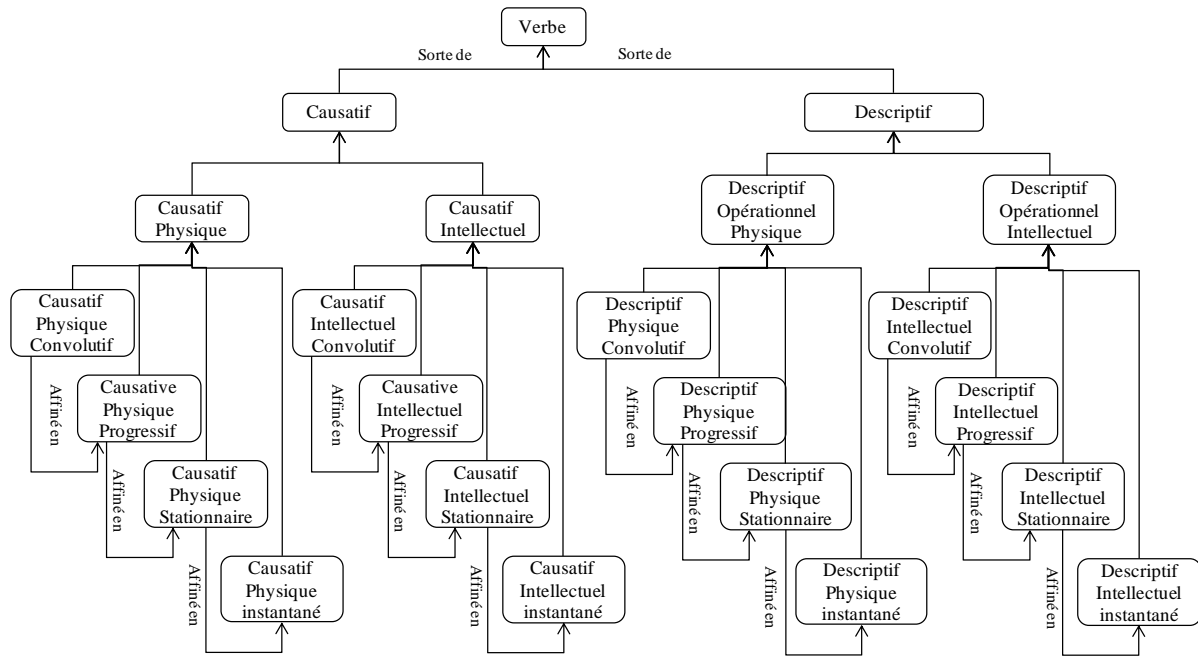


Figure 3.5. Ontologie de verbes

4.3. L'ontologie de domaine *pOnto*

L'ontologie des produits de domaine décrit les différents objets qui peuvent être utilisés. Elle définit une classification des éléments d'un domaine. Cette classification comporte, selon l'ontologie utilisée, plusieurs types d'éléments. Comme exemple, nous présentons l'ontologie des produits dans le domaine de la santé que nous avons utilisée. Dans cette ontologie nous distinguons cinq types d'éléments :

Les organisations. Une organisation correspond à une organisation médicale. Dans l'ontologie des produits, les types d'organisations médicales retenues sont : les unités de secours, les hôpitaux, les hôpitaux mobiles de l'armée et les infirmeries.

Les personnes. Ils correspondent à des personnes qui travaillent dans les organisations médicales. On distingue trois types de personnes : des employées, des médecins et des patients. Par exemple, un patient correspond à un blessé ou à un malade (les spécialisations). Les liens sont en général, de type « *isA* » ou de types spécifiques au domaine (« *supervisedBy* », « *worksIn* »...).

Les actes. Un acte médical correspond à un traitement, diagnostique... dans l'organisation médicale.

Les maladies. Les maladies sont classifiées selon la classification internationale des maladies.

Les symptômes. Les symptômes aussi sont classifiés selon la classification internationale des symptômes.

Un exemple de l'ontologie de santé est représenté dans la figure 3.6.

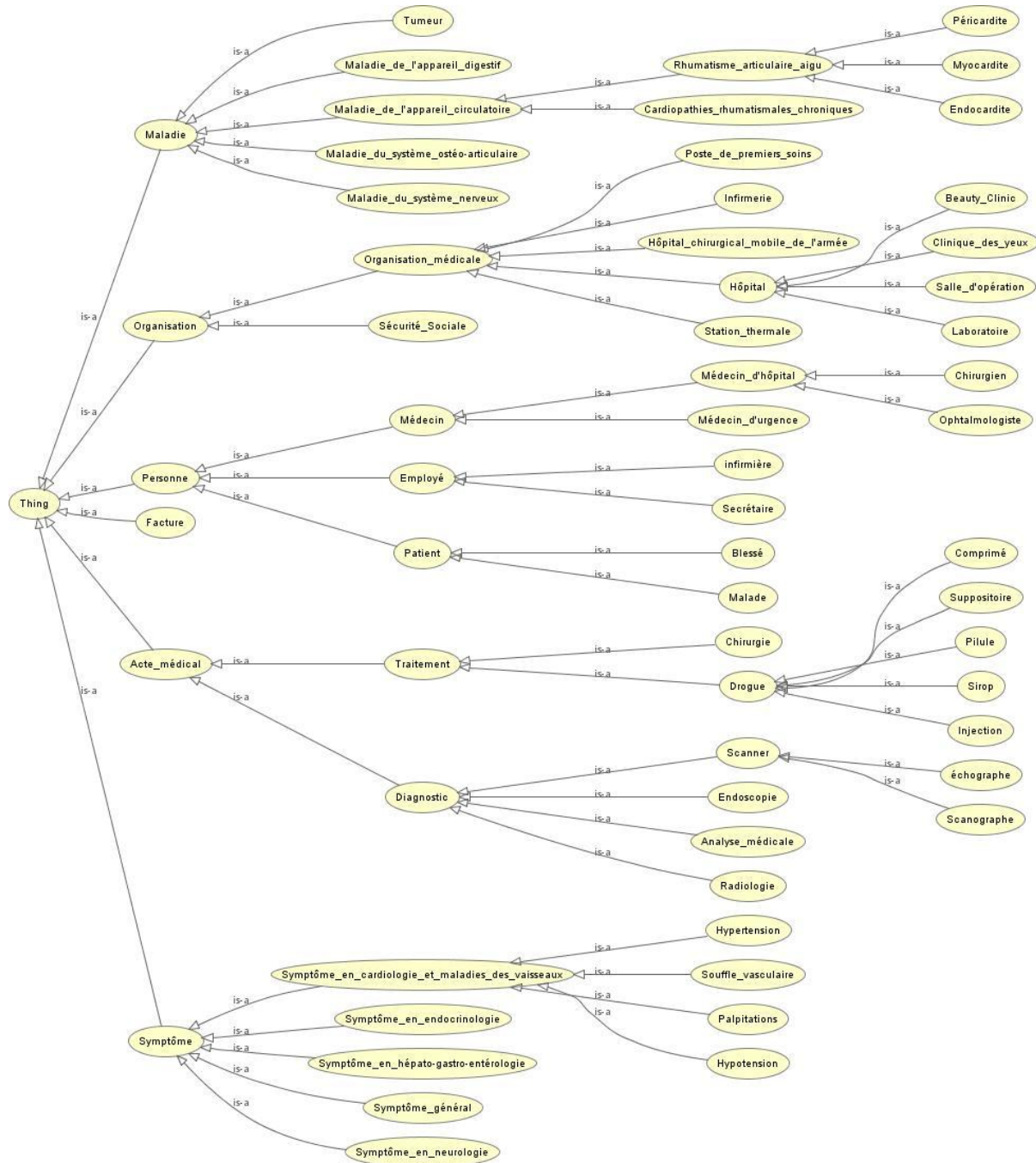


Figure 3.6. Exemple d'ontologie de domaine de la santé

4.4. Relation entre *vOnto* et *pOnto*

L'ontologie des verbes et l'ontologie des produits représentent les éléments des buts que l'on peut rencontrer dans le descripteur du service intentionnel et dans la requête de l'utilisateur. Dans cette approche, la description des buts est basée en partie sur la formalisation proposée dans [Prat, 1997]. Dans cette formalisation un but comporte un verbe, une cible et les paramètres (qui sont des concepts dans *pOnto*).

La relation entre ces deux ontologies est nécessaire pour la résolution des requêtes. Afin d'établir cette relation, nous rapportons chaque concept dans *pOnto* est lié à un ou aux plusieurs verbes dans *vOnto*.

Dans la figure 3.7, une image de l'ontologie de santé est indiquée lorsque un concept *Personne* de cette ontologie des produits (une ontologie de domaine) est liée à un autre concept *recevoir* dans l'ontologie des verbes. Le concept *recevoir* représente le verbe qui est utilisé par le concept de produit *Personne*. Nous pouvons remarquer que le verbe *recevoir* est un membre de la classe verbe « *verbes causatifs intellectuels* » dans l'ontologie des verbes, où nous pouvons trouver des verbes tels que *accueillir*, *recevoir* ...

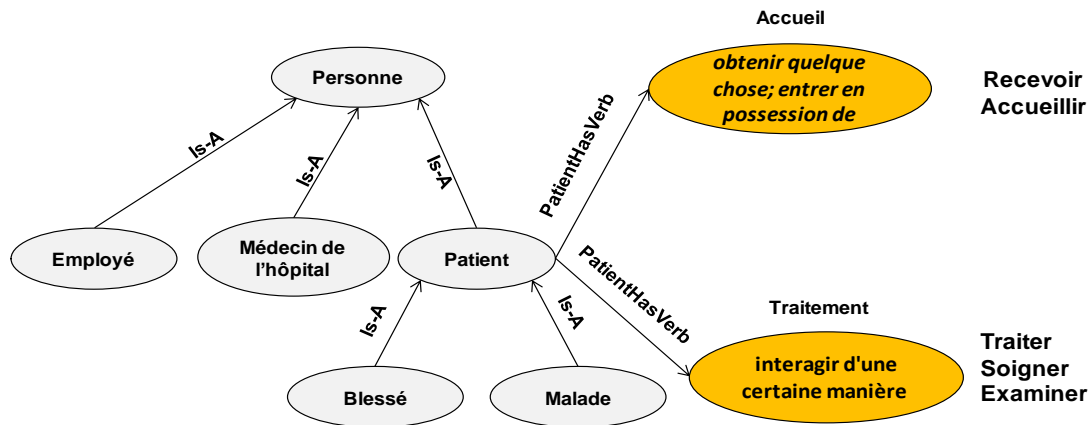


Figure 3.7. Relation entre *vOnto* et *pOnto*

Cette relation ajoute plusieurs informations sémantiques dans l'interprétation des requêtes des utilisateurs.

5. EXTENSION SEMANTIQUE DU DESCRIPTEUR DE SERVICE

Cette section détaille comment le descripteur du service intentionnel est implémenté dans l'annuaire étendu. L'annuaire étendu doit être capable de stocker des descripteurs de services de différentes formes (standard et intentionnel). Pour cette raison, l'annotation sémantique est utilisée comme technique d'extension sémantique d'un descripteur. L'extension sémantique du descripteur de service que nous définissons est basée sur un standard. Nous avons choisi d'étendre le langage SAWSDL pour décrire les services intentionnels. Cette extension vient combler un manque au niveau de la spécification de la nature de l'annotation sémantique dans SAWSDL. Pour cette raison, nous avons utilisé une ontologie dite « *technique* » que nous l'appelons *iSOnto* (l'Ontologie de services intentionnels), pour expliciter non seulement les concepts d'un domaine mais également la nature de ces concepts. La description est enrichie par des références portées sur des concepts techniques des services intentionnels tels que *but*, *verbe*, *produit*, *service atomique*, etc. Nous donnons un exemple de description des services dans la section 5.4.

5.1. L'usage de l'annotation

Une solution pour implémenter les descripteurs intentionnels est d'annoter un descripteur de base sémantiquement. L'annotation est un mécanisme d'extension très largement utilisé en génie logiciel pour ajouter de la connaissance sur des composants dans l'objectif est d'être interprétée et exploitée par un outil. Cette technique est également utilisée pour ajouter des propriétés sémantiques aux descripteurs dans le domaine des services web sémantiques.

Une annotation se structure en un couple <nom, valeur>. Le nom est spécifique au modèle sémantique supporté et la valeur est définie selon un domaine qui peut aller d'un domaine simple à une référence à un concept défini dans une ontologie de domaine.

On appelle annotation sémantique tout couple <nom, valeur> dont la partie valeur référence un concept dans une ontologie de domaine. Cette dernière technique est très largement utilisée dans le domaine des services [Paolucci et *al.*, 2002b] [Li et Horracks, 2003.] [Mandell et McIlraith, 2003] [Sycara et *al.*, 2003]; Plusieurs langages ont été définies tels que OWL-S [OWLS, 2003], WSML [de Bruijn et *al.*, 2005], SWORD [Ponnekanti et Fox, 2002] COSMOS [Fujii et Sudo, 2004], DAML + OIL [JointUS / UE, 2001] et les extensions de [Sivashanmugam et *al.*, 2003] pour ajouter de la sémantique à WSDL et UDDI l'aide d'ontologies DAML + OIL.

Ces approches utilisent des modèles sémantiques (ontologies...) pour annoter les descriptions des services web sémantiquement. Dans l'annotation utilisée, on peut référencer plusieurs concepts d'une ontologie de domaine, mais il n'y aurait aucune spécification de la nature des informations sémantiques puisque on ne distingue pas la nature des composants, et par conséquence, on perd les informations liées aux relations entre les concepts. La section suivante explique comment cette limitation est résolue dans l'approche YASA par l'introduction d'une ontologie technique.

5.2. L'approche YASA

Cette section résume l'approche YASA [Chabeb et Tata, 2008] [Chabeb et *al.*, 2009] pour étendre la description de domaine dans la description du service avec l'utilisation d'une ontologie technique (dans notre cas, cette ontologie technique est *iSOnto*). Dans YASA, la description de service est structurée en deux couches:

- Une description standard de base (OWL, WSDL...).
- Des descriptions techniques et de domaines représentées dans notre cas par des annotations en format XML.

L'amélioration proposée consiste à définir une annotation sémantique par un couple <nom, valeur> où le nom est une référence à une ontologie technique et la valeur est une référence à un concept d'ontologie de domaine. Cette nouvelle forme d'annotation permet de garantir que l'annuaire étendu est indépendant de la sémantique utilisée au langage de représentation. Par conséquent, plusieurs formes de descripteurs pourront être stockées et recherchées.

L'approche YASA étend sémantiquement et de manière générique un annuaire de type SAWSDL. En outre, SAWSDL ne permettant pas la description de comportement des services Web. [Chabeb et Tata, 2008] propose de décrire ce comportement à l'aide d'annotation sémantique selon une ontologie technique de service.

A l'instar de YASA, nous proposons d'utiliser une ontologie technique basée sur le modèle MiS (voir la figure 3.3) représentant l'ontologie *iSOnto*. Dans notre approche, nous faisons cette extension uniquement dans l'attribut « *interface* », figure 3.8.

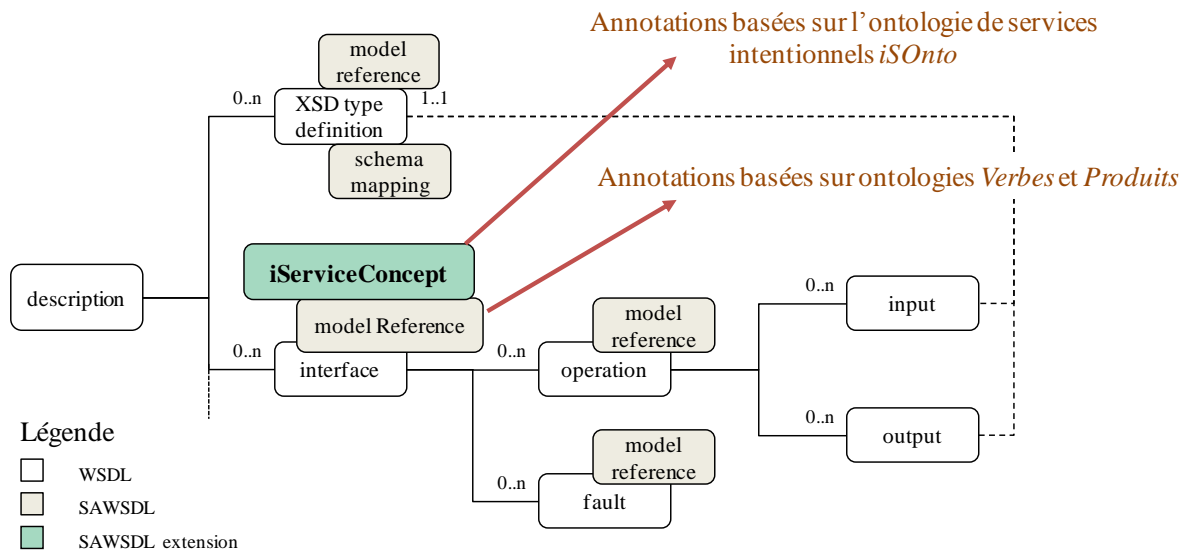


Figure 3.8. Structure du descripteur

5.3. Le descripteur intentionnel

Nous proposons d'ajouter dans le système d'annotation des descriptions de service Web SAWSDL, un nouvel attribut nommé *iServiceConcept*, Figure 3.9. Cet attribut permet l'annotation de plusieurs aspects d'un élément WSDL en utilisant des concepts techniques de l'ontologie de services intentionnels *iSOnto* qui correspondent des concepts d'une ontologie d'un domaine métier référencés dans l'attribut « *modelReference* » de SAWSDL [Aljoumaa *et al.*, 2010].

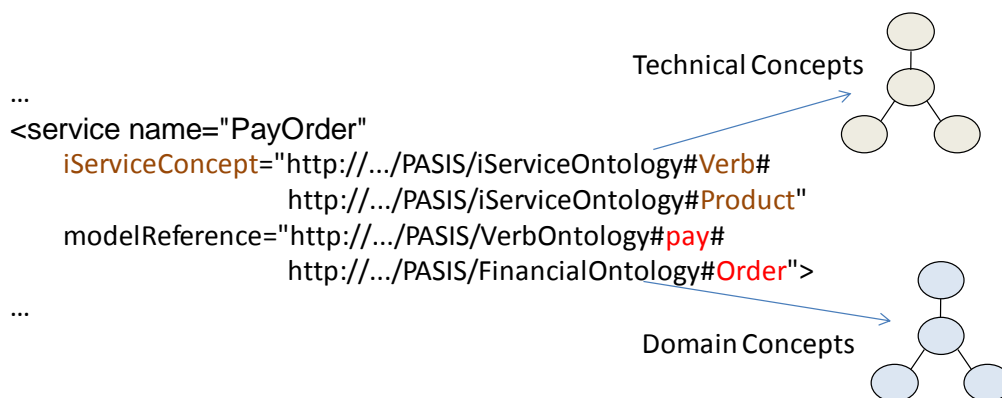


Figure 3.9. L'annotation dans le descripteur intentionnel

Dans l'interface du service *PayOrder*, son attribut *iServiceConcept* fournit une liste de concepts qui correspond au but, dans le même ordre, une liste de références du modèle de l'ontologie des verbes et de l'ontologie de domaine touristique fourni par l'attribut

modelReference, respectivement. C'est-à-dire que *verbe* est associé au concept de *pay* et le produit est associé au concept de *order*.

Un avantage à souligner dans cette approche est que nous concevons une ontologie technique de services intentionnels avec la possibilité de l'étendre à tout moment par de nouveaux concepts. Ainsi, ces extensions et ces enrichissements n'auront aucun impact sur le système d'annotation [Chabeb *et al.*, 2009]. Cela permet de maintenir le système de l'annotation indépendante de l'ontologie de domaine utilisée.

Cependant, cette modification implique la modification de XSD de SAWSDL afin qu'il accepte le nouvel attribut. Dans le tableau 3.1, nous présentons le XSD modifié de SAWSDL. Nous introduisons la ligne ajoutée en bleu gras.

Tableau 3.1. XSD de SAWSDL étendu

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  CibleNamespace="http://www.w3.org/ns/sawsdl"
  xmlns="http://www.w3.org/ns/sawsdl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:wsdl="http://www.w3.org/ns/wsdl">

  <xs:simpleType name="listOfAnyURI">
    <xs:list itemType="xs:anyURI"/>
  </xs:simpleType>

  <xs:attribute name="iServiceConcept" type="listOfAnyURI" />
  <xs:attribute name="modelReference" type="listOfAnyURI" />
  <xs:attribute name="liftingSchemaMapping" type="listOfAnyURI" />
  <xs:attribute name="loweringSchemaMapping" type="listOfAnyURI" />

  <xs:element name="attrExtensions">
    <xs:complexType>
      <xs:annotation>
        <xs:documentation>This element is for use in WSDL 1.1 only. It
          does not apply to WSDL 2.0 documents. Use in
          WSDL 2.0 documents is invalid.</xs:documentation>
      </xs:annotation>
      <xs:anyAttribute namespace="##any" processContents="lax" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Le tableau 3.2, présente l'annotation sémantique d'un service intentionnel nommé *OrderRequest*. L'attribut *modelReference* référence des concepts de l'ontologie de domaine : *FinancialOntology*. L'attribut *iServiceConcept* offre la capacité de distinguer les rôles joués par les concepts *order*, *Request*, *by periodic planning*, *OrderRequest*, *FillRequest* et *SendRequest* référencés par l'attribut *modelReference*. Pour cela, *iServiceConcept* spécifie que ces concepts correspondent respectivement aux concepts de services : *Verb*, *Product*, *Manner*, *Sequence*, *Service Component* et *Service Component*. L'ordre est important, puisque on associe le premier concept technique « *verb* » au premier concept de domaine « *Order* », le second concept technique « *Product* » au second concept de domaine « *Request* », et ainsi de suite.

Tableau 3.2. Exemple d'un service intentionnel présenté dans SAWSDL

```

<wsdl:description
  CibleNamespace="http://www.w3.org/2002/ws/sawSDL/spec/wsdl/order#"
  xmlns="http://www.w3.org/2002/ws/sawSDL/spec/wsdl/order#"
  xmlns:wsdl="http://www.w3.org/ns/wsdl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sawSDL="http://www.w3.org/ns/sawSDL">

  <wsdl:types>
    <xs:schema
      CibleNamespace="http://www.w3.org/2002/ws/sawSDL/spec/wsdl/order#"
      elementFormDefault="qualified">
        <xs:element name="OrderRequest"

sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purch
aseorder#OrderRequest"

sawSDL:loweringSchemaMapping="http://www.w3.org/2002/ws/sawSDL/spec/mapping
/RDFOnt2Request.xml">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="customerNo" type="xs:integer" />
          <xs:element name="orderItem" type="item" minOccurs="1"
maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:complexType name="item">
      <xs:all>
        <xs:element name="UPC" type="xs:string" />
      </xs:all>
      <xs:attribute name="quantity" type="xs:integer" />
    </xs:complexType>
    <xs:element name="OrderResponse" type="confirmation" />
    <xs:simpleType name="confirmation"

sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purch
aseorder#OrderConfirmation">
      <xs:restriction base="xs:string">
        <xs:enumeration value="Confirmed" />
        <xs:enumeration value="Pending" />
        <xs:enumeration value="Rejected" />
      </xs:restriction>
    </xs:simpleType>
  </xs:schema>
</wsdl:types>

  <wsdl:interface name="Order"

sawSDL:iServiceConcept="&http://.../iServiceOntology;#verb
&http://.../iServiceOntology;#product
&http://.../iServiceOntology;#manner
&http://.../iServiceOntology;#sequence
&http://.../iServiceOntology;#serviceComponent
&http://.../iServiceOntology;#serviceComponent ">
sawSDL:modelReference="&http://.../VerbOntology;#order
&http://.../purchaseOntology;#request

```

```

&http://.../purchaseOntology;#by Periodic Planning
&http://.../iServiceOntology;#OrderRequest
&http://.../iServiceOntology;#FillRequest
&http://.../iServiceOntology;#SendRequest">

<wsdl:operation name="order" pattern="http://www.w3.org/ns/wsdl/in-out"
sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purch
aseOrder#RequestPurchaseOrder">
  <wsdl:input element="OrderRequest" />
  <wsdl:output element="OrderResponse" />
</wsdl:operation>
</wsdl:interface>
</wsdl:description>

```

5.4. Exemple du descripteur intentionnel

Notre exemple de [Aljoumaa et al., 2011b], montre comment un fournisseur peut publier les services intentionnels du moment de leur découverte à partir des besoins des utilisateurs et jusqu'au moment de leur publication dans l'annuaire. Cet exemple est extrait d'une étude de cas réel de la gestion de l'approvisionnement d'électricité ESM (Electricity Supply Management) [Nurcan et Rolland, 1999] pour illustrer la publication de services. Il est représenté en utilisant le formalisme de la carte.

Cet exemple concerne la vente de l'électricité aux utilisateurs finaux. Par souci de concision, nous présentons trois services intentionnels d'ESM. Dans le Tableau 3.3,

Tableau 3.3. *Trois services d'ESM*

Type	Service
Atomique	S Measure electricity by remote meter reading
Agrégat à variation	S Get financial counterpart
Agrégat composite	S Sell electricity with credit strategy

Les buts des trois services sont formulés au Tableau 3.4

Tableau 3.4. *La formulation des buts de services*

Goals
$(Measure)_{verb} (electricity)_{product} (by meter reading)_{manner}$
$(Get)_{verb} (financial counterpart)_{result}$
$(Sell)_{verb} (electricity)_{product} (with credit strategy)_{means}$

5.4.1. Représentation des services intentionnels dans *iSOnto*

Avant de passer à la mise en œuvre du descripteur, nous présentons les concepts de services liés à l'ontologie de services intentionnels. La Figure 3.10, introduit la présentation du service atomique $S_{Measure electricity by remote meter reading}$ (S1) dans *iSOnto*. Comme on peut le voir sur cette

figure, nous avons fait une distinction graphique entre les nœuds et les feuilles lors de l'instanciation de l'ontologie *iSOnto*.

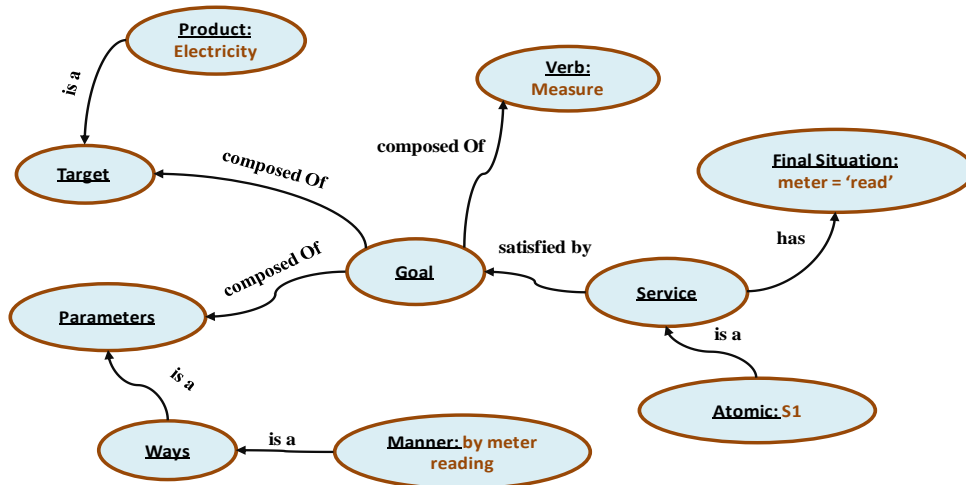


Figure 3.10. Présentation du service atomique $S_{\text{Measure electricity by remote meter reading}}$ dans *iSOnto*

Dans la Figure 3.11, nous introduisons la présentation du service à variation $S_{\text{Get financial counterpart}}$ (S9). Ce service agrégat spécifie deux alternatives pour obtenir de la contrepartie financière : soit par « *contract* » ou par « *consumption* ».

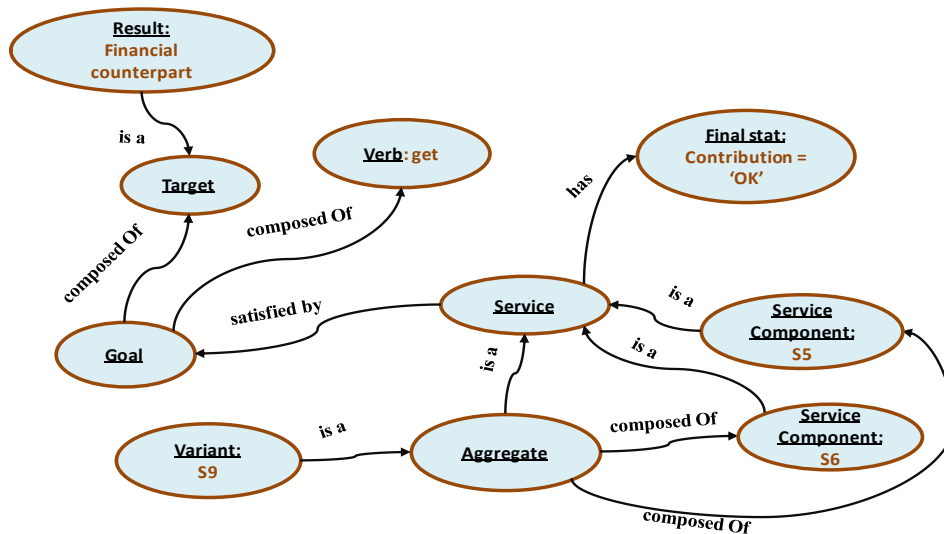


Figure 3.11. Présentation du service à variation $S_{\text{Get financial counterpart}}$ dans *iSOnto*

Et dans la Figure 3.12, nous introduisons la présentation du service composite $S_{\text{Sell Electricity with credit strategy}}$ (S10). Ce service intentionnel agrégat est composé d'une séquence de trois services intentionnels : deux services à variation ($S_{\text{Measure electricity consumption}}$, $S_{\text{Get financial counterpart}}$) et un service intentionnel atomique ($S_{\text{Recover financial counterpart}}$).

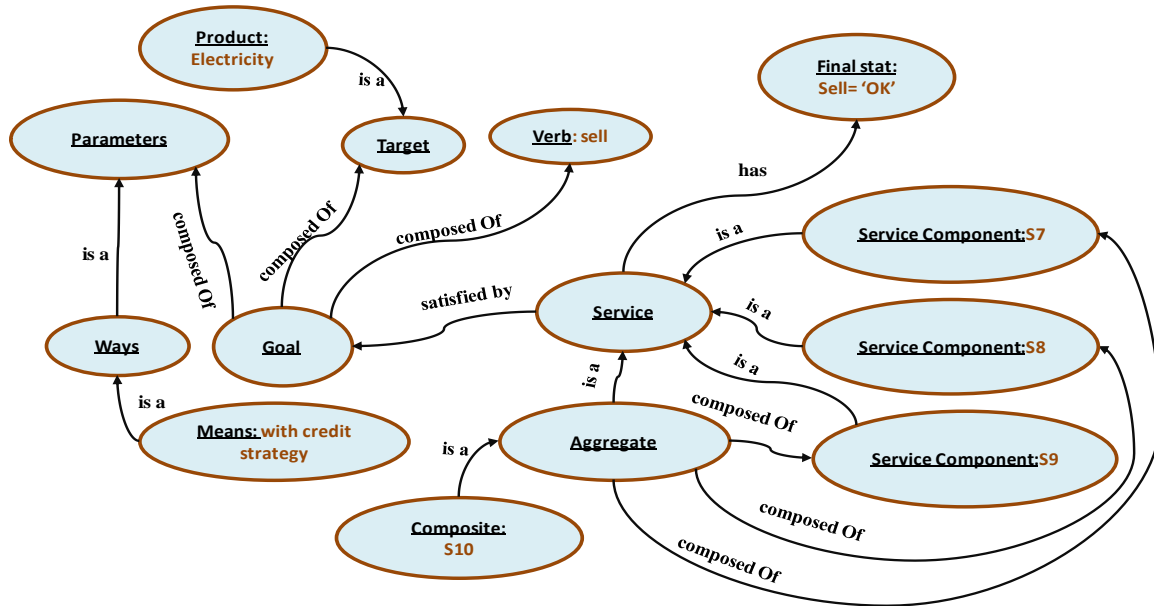


Figure 3.12. Présentation du service composite $S_{\text{Sell Electricity with credit strategy}}$ dans *iSOnto*

Le Tableau 3.5 contient les trois précédents services : atomique $S_{\text{Measure electricity by remote meter reading}}$, composite $S_{\text{Sell Electricity with credit strategy}}$ et à variation $S_{\text{Get financial counterpart}}$ en utilisant les concepts d'*iSOnto*, de la même manière qu'ils vont être présentés dans le descripteur.

Tableau 3.5. Présentation des descriptions dans *iSOnto*

$(\text{Measure})_{\text{verb}}$ $(\text{electricity})_{\text{product}}$ $(\text{by meter reading})_{\text{manner}}$ $(\text{Measure electricity by meter reading})_{\text{atomic}}$
$(\text{Get})_{\text{verb}}$ $(\text{financial counterpart})_{\text{result}}$ $(\text{Get financial counterpart})_{\text{variant}}$ $(\text{Get Financial counterpart Based On Contract})_{\text{Service Component}}$ $(\text{Get Financial counterpart Based On Consumption})_{\text{Service Component}}$
$(\text{Sell})_{\text{verb}}$ $(\text{electricity})_{\text{product}}$ $(\text{with credit strategy})_{\text{means}}$ $(\text{Sell electricity with credit strategy})_{\text{composite}}$ $(\text{Measure electricity})_{\text{Service Component}}$ $(\text{Get Financial counterpart})_{\text{Service Component}}$ $(\text{Cover Financial counterpart})_{\text{Service Component}}$

5.4.2. Descripteurs des services intentionnels en SAWSDL

Nous présentons uniquement la partie modifiée du descripteur en utilisant SAWSDL afin de présenter l'annotation ajoutée aux services dans le Tableau 3.5. Par conséquent, le descripteur du service atomique $S_{\text{Measure electricity by remote meter reading}}$ est présenté dans le Tableau 3.6.

Tableau 3.6. Le descripteur du service atomique $S_{\text{Measure electricity by remote meter reading}}$

```
<service name="MeasureElectricityByMeterReading"
```

```
iServiceConcept="http://.../iServiceOntology#Verb#product#manner#Atomic"
modelReference="http://.../VerbOntology#measure#/OntologyElectricity
#electricity#/OntologyElectricity#meter Reading " >
```

Le descripteur du service variante à variation $S_{Get\ financial\ counterpart}$ est présenté dans le Tableau 3.7.

Tableau 3.7. Le descripteur du service à variation $S_{Get\ financial\ counterpart}$

```
<service name="GetFinancialcounterpart"
  iServiceConcept="http://.../iServiceOntology#Verb#Result#Variant
#ServiceComponent#ServiceComponent"
modelReference="http://.../VerbOntology#get
#OntologyElectricity#Financialcounterpart
#OntologyElectricity#
#OntologyElectricity#GetFinancialcounterpartBasedOnContract
#OntologyElectricity#GetFinancialcounterpartBasedOnConsumption" >
```

Enfin, le descripteur du service composite $S_{Sell\ Electricity\ with\ credit\ strategy}$ est présenté dans le Tableau 3.8.

Tableau 3.8. Le descripteur du service composite $S_{Sell\ Electricity\ with\ credit\ strategy}$

```
<service name="SellElectricityWithCreditStrategy"
iServiceConcept="http://.../iServiceOntology#Verb#product#manner
#Composite#ServiceComponent#ServiceComponent#ServiceComponent"
modelReference="http://.../VerbOntology#Sell
#OntologyElectricity#Electricity
#OntologyElectricity# creditStrategy#
#OntologyElectricity#MeasureElectricity
#OntologyElectricity#GetFinancialcounterpart
#OntologyElectricity# Cover Financial counterpart" >
```

6. MODELE DE CARTE

Le modèle de Carte est présenté dans cette section, parce qu'il est utilisé pour formaliser les processus de guidage méthodologique proposés dans cette thèse pour publier et rechercher des services intentionnels (chapitres 4 et 5). Cette utilisation de Carte a été décrite dans [Ralyté, 2001]. Nous en faisons ici un rappel.

Le modèle de Carte est utilisé dans le domaine des processus d'ingénierie pour mettre en avant les intentions et les stratégies possibles pour atteindre celles-ci. Il permet ainsi de fournir un ensemble de prescriptions à l'utilisateur et de l'aider à sélectionner de manière dynamique la prescription la plus adaptée en fonction de la situation du produit rencontrée. Ce modèle est un modèle de processus multi-démarches permettant la construction dynamique du processus. Ces prescriptions reposent sur l'idée de progression intentionnelle que l'on peut exprimer à travers les deux faits suivants :

- Toute transformation du produit en cours de développement est la concrétisation d'une intention de l'utilisateur
- Chaque nouvelle transformation du produit résulte d'une progression à partir d'une intention déjà réalisée.

Le modèle de Carte introduit de la flexibilité dans la construction des prescriptions de processus. Il a l'avantage d'être adapté au raisonnement intentionnel de l'utilisateur qui l'applique. A aucun moment il n'est forcé de réaliser une intention particulière ou d'appliquer une stratégie de réalisation d'intention spécifique, à moins que la situation ne l'exige.

Les avantages de l'utilisation du Modèle de Carte sont :

- La reconnaissance explicite du rôle des stratégies dans la modélisation de processus ;
- Un modèle non-prescriptif d'intentions et de stratégies contenant des alternatives à partir desquelles le processus réel peut être construit ;
- La construction dynamique du processus est la règle plutôt qu'une exception.

6.1. Méta-modèle de processus

La Figure 3.13 montre le modèle de Carte en tant que méta-modèle de processus (Map). Une Carte est composée d'un ordonnancement non figé d'intentions et de stratégies (le modèle duplique l'entité « intention » pour des raisons de lisibilité. Les deux entités représentent un seul concept).

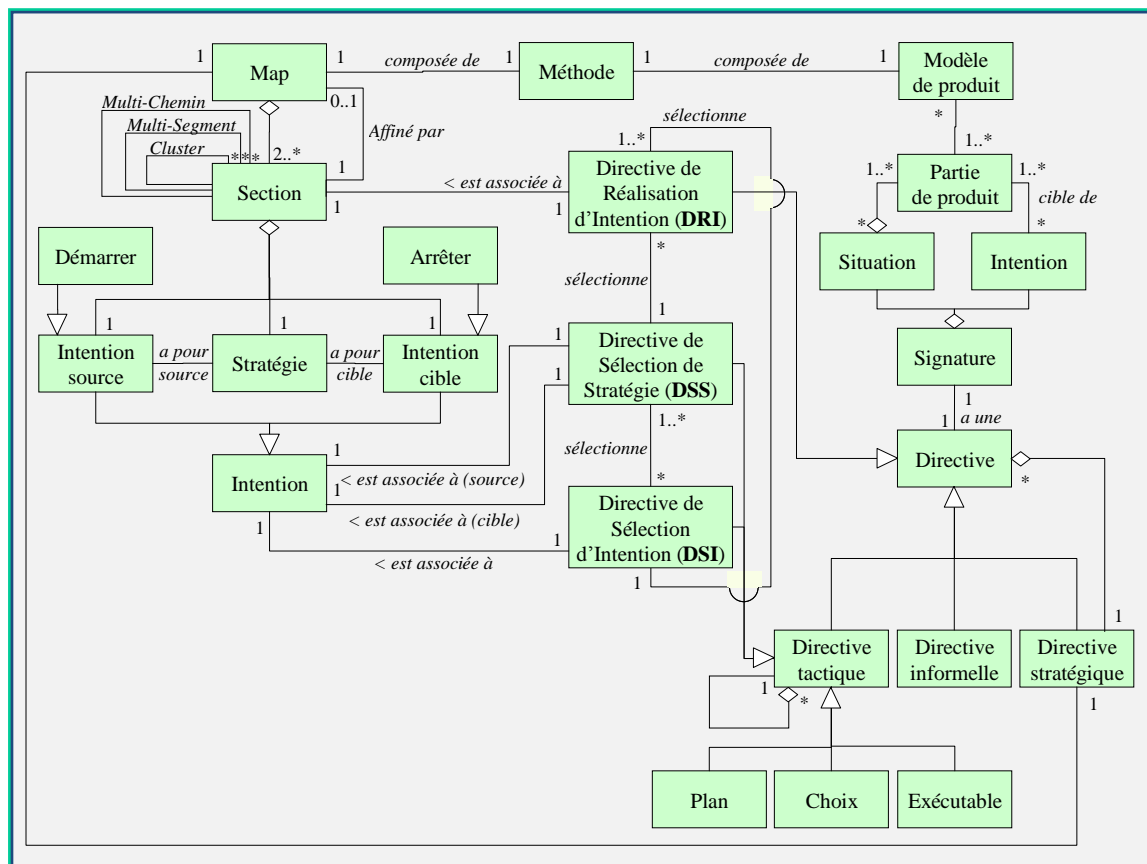


Figure 3.13. Le modèle de la Carte

Dans une carte, chaque section est décrite au niveau type et peut de ce fait être *exécutée* plusieurs fois. Par « *exécuter une section* $\langle I_s, I_c, S \rangle$ », nous entendons réaliser l'intention cible I_c avec la stratégie S en exploitant le résultat d'une réalisation précédente de l'intention source I_s . En fait, dès lors que l'intention source a été réalisée au moins une fois, l'intention cible peut être réalisée autant de fois que nécessaire, en utilisant la stratégie S et le résultat de la réalisation de I_s .

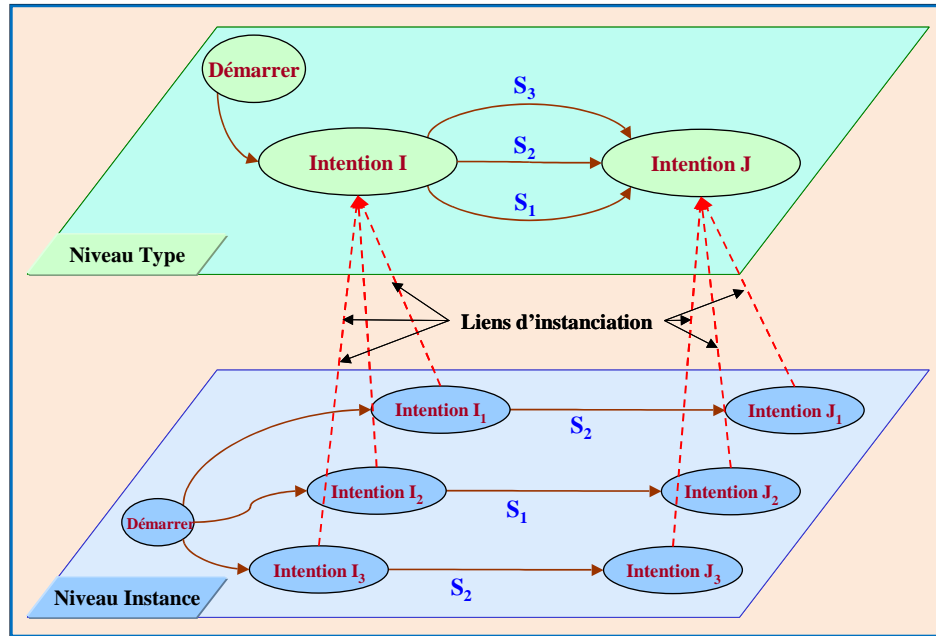


Figure 3.14. Les niveaux type et instance de la carte

Donc, une carte est un processus exprimé au niveau « type ». Une section peut être exécutée plusieurs fois. De plus, la même intention peut être réalisée plusieurs fois sur des parties de produits différentes. Ceci est illustré à la Figure 3.14 où I et J sont des intentions au niveau type, et I_1 - I_3 et J_1 - J_3 sont différentes instances des intentions I et J .

L'exécution d'une carte commence en sélectionnant une section dont l'intention source est *Démarrer*. La réalisation d'une intention se fait à l'aide d'une stratégie qui exploite le résultat des sections déjà exécutées. Comme l'intention *Démarrer* ne peut être cible d'aucune section, la réalisation de toute autre intention ne sera pas guidée si elle n'est pas cible d'une section. Nous considérons que l'exécution d'une carte commence au moment où l'utilisateur a déjà démarré, mais n'a pas encore réalisé une autre intention de la carte.

Arrêter est la seule intention qui n'est source d'aucune section. Le choix d'une section dont l'intention cible est *Arrêter* termine l'exécution de la carte.

6.2. Principe de guidage

Dans un processus dirigé par les intentions, l'utilisateur est confronté de manière répétitive aux deux questions suivantes :

- Comment réaliser l'intention sélectionnée ?
- Comment sélectionner la prochaine intention afin de progresser dans le processus ?

Le guidage est le support documenté qui permet à un utilisateur d'orienter ses choix lors de l'exécution d'une Carte. Pour guider l'utilisateur dans la construction dynamique du chemin qu'il veut suivre, des « directives » sont associées à la carte. Ces directives sont appliquées de manière situationnelle qui oriente la sélection de la directive suivante à exécuter. Ainsi une situation qui est exprimée par l'ensemble des états des produits manipulés par une Carte est pertinente pour une directive.

Dans les sous-sections suivantes, nous présentons la notion de directive, la typologie des directives et enfin nous détaillons les directives associées au modèle de Carte (DSS, DSI et DRI).

6.2.1. Notion de directive

Une directive est définie comme « un ensemble d'indications sur la façon de procéder pour réaliser un objectif ou exécuter une activité » [Le Petit Robert, 2000].

La première option, qui est celle relative à la réalisation d'un objectif, s'applique dans notre cas. Plus précisément, selon [Ralyté, 2001] une directive définit la connaissance du domaine pour guider l'utilisateur dans la réalisation d'une intention dans une situation donnée. Elle préconise un processus à suivre pour réaliser une intention.

Comme le montre le méta-modèle de processus Carte (Figure 3.13), une directive a une *signature* et un *corps*. La *signature* représente la partie visible de la directive. Elle caractérise les conditions dans lesquelles celle-ci peut être appliquée et le résultat qu'elle permet d'obtenir, sans pour autant dire comment faire. Le *corps* définit la démarche à suivre pour satisfaire l'intention capturée dans la signature. Il fournit des conseils qui guident l'exécution du processus de la directive et le relie avec les parties de produit impliquées.

Dans les sous-sections suivantes nous détaillons les notions de *signature* et de *corps* d'une directive.

6.2.2. Signature

Le méta-modèle montre que la *signature* d'une directive est définie par un couple $\langle (situation), intention \rangle$. Les conditions d'application de la directive sont capturées dans sa situation. Le résultat auquel elle permet d'aboutir est capturé dans son intention. Par conséquent, chaque directive s'applique dans une situation particulière pour satisfaire une intention particulière. Le corps de cette directive contient les recommandations pour atteindre cet objectif.

Puisque le processus à réaliser pour satisfaire l'intention de la directive est capturé dans le corps de la directive, on peut voir une directive comme une boîte noire qui encapsule le processus de transformation du produit d'entrée en produit de sortie comme le montre la Figure 3.15.

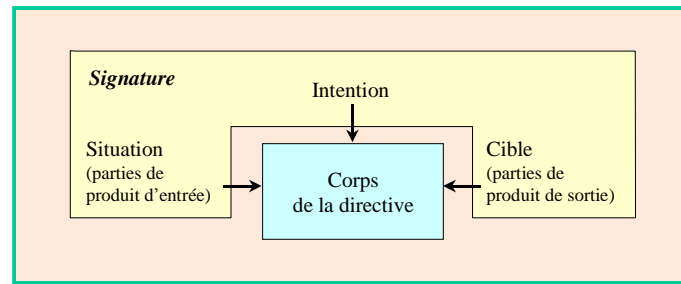


Figure 3.15. *Vue générale d'une directive*

Le produit d'entrée représente la situation dans laquelle la directive peut être appliquée. Il peut être composé de plusieurs parties du produit en construction qui sont nécessaires pour commencer l'exécution du processus encapsulé dans le corps de la directive. L'intention reflète le but à atteindre dans cette situation. Le produit de sortie représente le résultat de l'exécution de la directive. C'est la cible de son intention.

Nous détaillons par la suite la *situation* et l'*intention* d'une signature de directive.

6.2.2.1. Situation d'une signature

La *situation* dans la signature d'une directive identifie une partie de produit en cours de développement nécessaire à la satisfaction de l'intention de la directive (Figure 3.13). Elle exprime la situation dans laquelle la directive peut être appliquée pour la satisfaction de l'intention qui lui est associée.

Chaque partie de produit référencée dans la situation est un élément du modèle de produit. Elle fait partie de l'ensemble des parties de produit associées à la directive. Il peut s'agir d'un élément de produit atomique, d'une association de plusieurs éléments de produit ou même du modèle de produit de la méthode en entier (une carte).

6.2.2.2. Intention d'une signature

L'*intention* de la directive est un but, un objectif qu'un utilisateur désire réaliser à un moment donné du processus. Elle exprime un but que l'utilisateur souhaite atteindre en appliquant la directive. Par exemple, « *Annoter les services* » est une des intentions que l'on peut exprimer dans le processus de la publication.

L'expression de l'intention suit une adaptation de la structure prédéfinie de but proposée par [Prat, 1997]. Le fait que la cible soit intégrée dans l'expression de l'intention permet de caractériser le processus de la directive par le couple <(situation), intention> appelé la signature de la directive.

La cible de l'intention est composée d'une ou plusieurs parties de produit. Dans la Figure 3.13, on a dupliqué le type *Intention* pour simplifier la présentation graphique du modèle.

6.2.3. Corps

Le *corps* d'une directive exprime explicitement le guidage fourni par celle-ci. Il contient un ensemble de recommandations définissant comment procéder pour satisfaire l'intention définie dans la signature de la directive. Il propose une ou plusieurs démarches à suivre pour

aboutir au résultat attendu. Le corps de la directive représente la partie interne alors que la signature représente l'interface de la directive.

6.3. Typologie des directives

Le méta-modèle de processus Carte (Figure 3.13) montre qu'il y a trois types de directives en fonction de leur complexité, de leur richesse et de la manière dont elles sont exprimées : *directive stratégique*, *directive tactique*, et *directive simple* [Ralyté, 2001].

- Une *directive stratégique* a une structure de graphe, représentée par une carte.
- Une *directive tactique* est une structure d'arbre. Elle est composée d'autres directives. Elle peut être de type *plan*, *choix* ou *exécutable*.
- Une *directive informelle* est une directive simple non structurée.

Dans la suite, nous présentons ces trois types de directive.

6.3.1. Directive stratégique

Une *directive stratégique* a une structure de graphe. Comme le montre la Figure 3.13, elle est représentée sous forme d'une carte et d'un ensemble de directives associées [Rolland et al., 1999], [Benjamin, 1999]. Elle représente une vue stratégique de la démarche de développement multi-démarche basée sur un ensemble d'intentions et de stratégies. La directive stratégique permet d'exprimer un processus en proposant plusieurs chemins possibles pour satisfaire son intention.

Comme on l'a vu à la Figure 3.13, trois types de directive (DRI, DSS et DSI) sont associés à la carte. Puisque la directive stratégique est représentée par une carte, alors, les trois types de directives lui sont associés.

Dans la section (6.4), nous présentons en détail les directives associées à la carte ou qui composent la directive stratégique représentée par cette carte.

6.3.2. Directive tactique

Une *directive tactique* a une structure d'arbre. C'est une directive composée d'autres directives, sous forme d'un *plan*, d'un *choix* de plusieurs sous-directives alternatives ou sous forme exécutable. Le formalisme utilisé dans ce type de directive est inspiré du formalisme de modélisation des processus NATURE [Rolland et al., 1995], [Jarke et al., 1999]. Ce dernier repose sur la notion du *contexte* et représente le processus de développement par une hiérarchie de contextes [Rolland et Prakash, 2000] [Plihon, 1994] [Plihon et Rolland, 1995]. La typologie de contextes proposée dans cette approche est appliquée dans notre méta-modèle et nous permet de spécialiser la directive tactique en trois types : *Plan*, *choix* et *exécutable*.

La Figure 3.16 montre les directives tactiques en faisant apparaître que celles-ci sont structurées en une hiérarchie de directives tactiques. Les trois types de directive tactique sont détaillés dans les sous-sections suivantes.

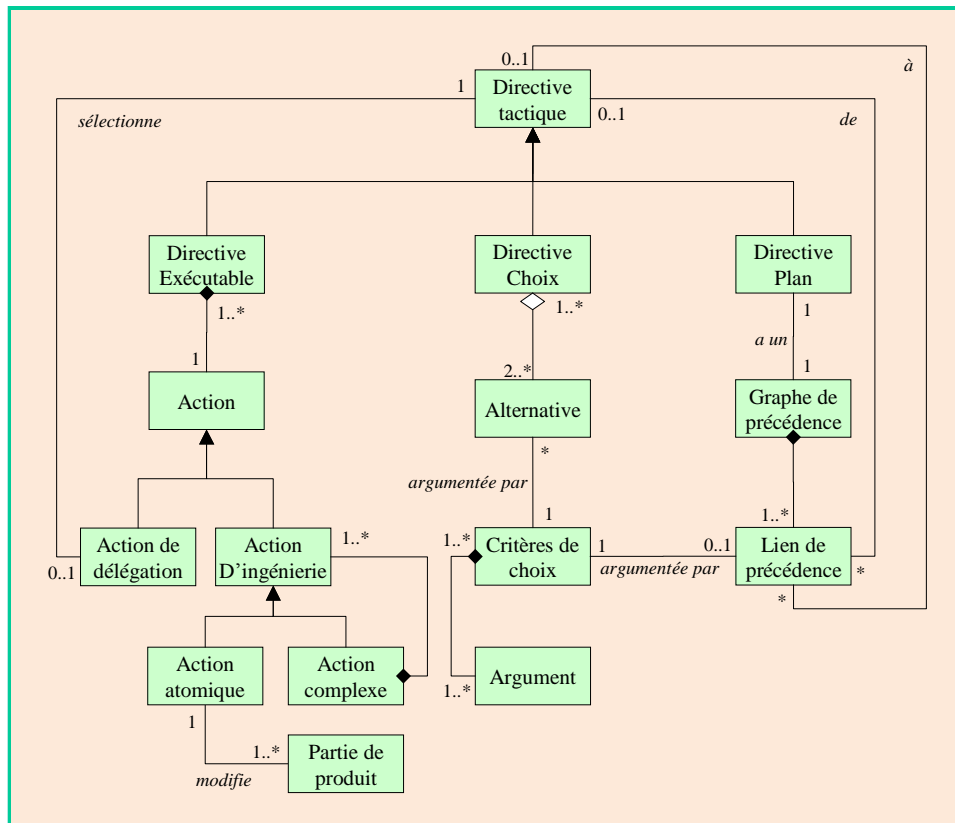


Figure 3.16. Structure d'une directive tactique

6.3.2.1. Directive choix

Une directive *choix* correspond à une situation qui nécessite l'exploration de différentes alternatives (Figure 3.17). Il y a des situations dans lesquelles l'utilisateur dispose de différentes façons d'atteindre le but qu'il poursuit. Il doit faire un choix parmi un ensemble de possibilités permettant de résoudre le problème. Chaque solution alternative est décrite par une nouvelle directive qui permet de satisfaire la même intention que celle de la directive choix. Ces directives alternatives peuvent appartenir à l'un des deux types possibles : tactique ou informelle.

La directive choix permet de décomposer une intention en sous-intentions alternatives et d'affiner ainsi l'intention de haut niveau en intentions plus fines. Les alternatives d'une directive choix précisent l'intention de cette dernière, soit en ajoutant une information sur l'approche suivie pour mettre en œuvre l'intention, soit en décrivant les différentes transformations du produit qui peuvent être menées pour atteindre l'objectif de la directive choix. Le lien hiérarchique entre les directives est appelé le *lien de choix*.

L'utilisateur explore les différentes possibilités pour la résolution d'un problème à l'aide des *critères de choix*. Ces critères aident à choisir l'alternative la mieux appropriée aux caractéristiques de la situation courante. Un *critère de choix* est une combinaison d'*arguments* en faveur ou en défaveur du choix d'une alternative. Les arguments reposent sur des heuristiques ou sur les caractéristiques du produit en cours de développement. Ils sont atomiques et peuvent être réutilisés dans plusieurs critères de choix. Ils sont décrits en langage naturel.

La Figure 3.17 propose un exemple d'une directive choix. Dans cet exemple, l'intention « *Annoter un verbe causatif physique* » est affinée par quatre alternatives proposant différents choix pour satisfaire cette intention : (1) « *Annoter un verbe causatif physique de service* », (2) « *Annoter un verbe causatif physique de processus* », (3) « *Annoter un verbe causatif physique d'activité* » ou (4) « *Annoter un verbe causatif physique d'action* ». On associe un argument à chaque alternative pour motiver le choix de cette alternative plutôt que les autres. Pour choisir la première alternative, l'argument (a1) (*Si le verbe est un verbe causatif physique de service*) doit être évalué et plus important que les autres arguments.

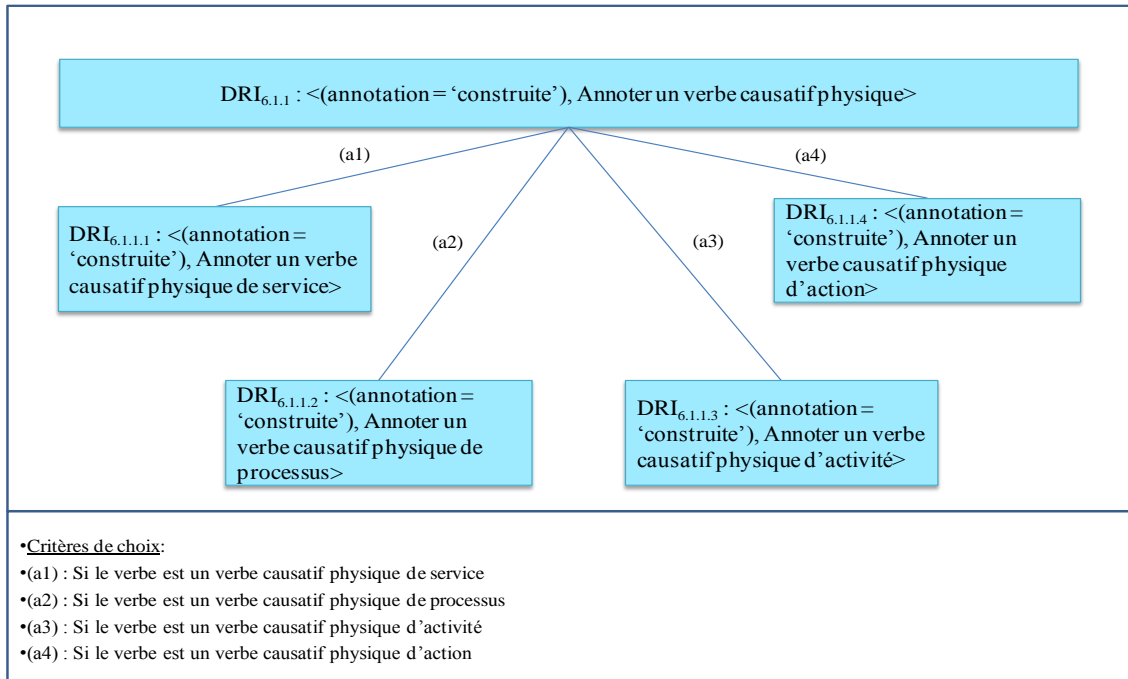


Figure 3.17. Exemple d'une directive choix

6.3.2.2. Directive plan

Une directive plan correspond à un problème complexe qui, pour être résolu, nécessite d'être décomposé en un ensemble de sous-problèmes. L'utilisateur connaît l'ensemble des décisions qui lui permettront d'atteindre le but qu'il poursuit. Il a un plan composé d'un ensemble de sous-directives. L'ordre d'exécution des directives composantes est défini dans le *graphe de précedence* (Figure 3.18). Il y a un graphe par directive plan. Les nœuds du graphe sont des directives (les composants du plan), alors que les arcs appelés *liens de précedence* représentent des transitions ordonnées ou parallèles entre directives.

Les *critères de choix* attachés aux liens permettent de prescrire les conditions d'occurrence d'une transition. Ils sont construits à partir d'arguments et sont de même nature que ceux attachés aux alternatives des contextes choix. Leur but est d'aider l'utilisateur à choisir le chemin à suivre pour l'exécution du plan.

Dans certains cas, le graphe de précedence peut être simple et ne proposer qu'un seul chemin pour exécuter le plan. Il est alors inutile d'associer des critères de choix aux liens de précedence.

Un graphe de précedence peut aussi contenir plusieurs chemins d'exécution. Ceci permet d'introduire plus de flexibilité dans la démarche capturée par la directive.

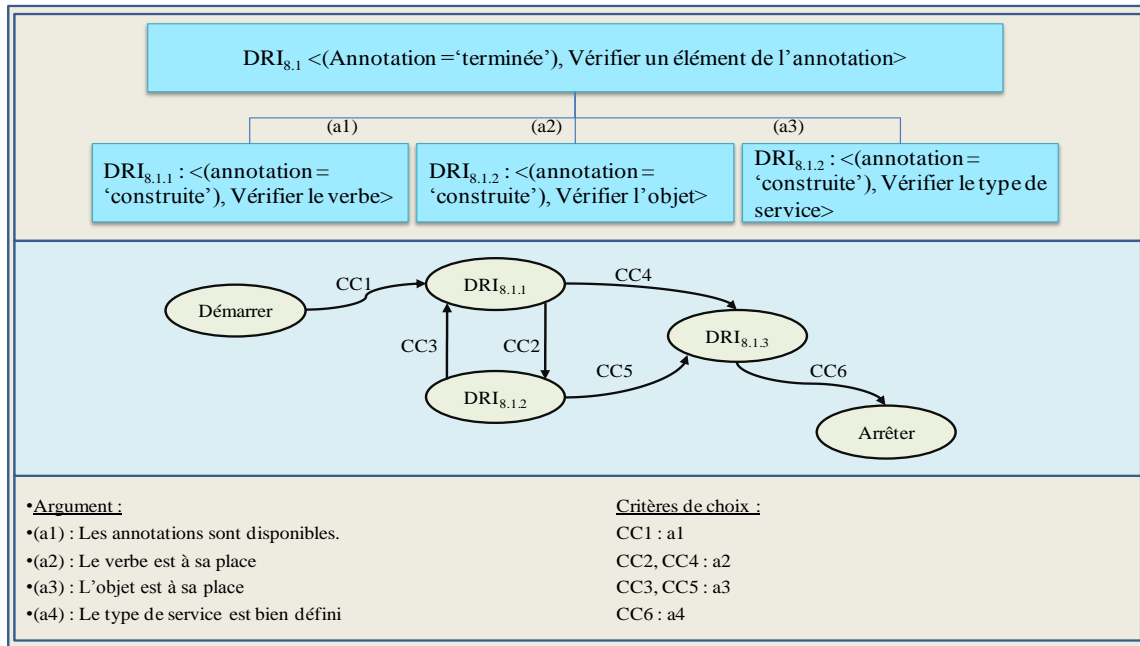


Figure 3.18. Exemple de directive plan

Exécuter un plan revient à parcourir son graphe de précedence. Lors de l'exécution d'un contexte plan, les critères de choix des liens de précedence sont calculés grâce à l'évaluation de leurs arguments.

6.3.2.3. Directive exécutable

Une directive exécutable correspond à une intention qui peut être concrétisée par une action de transformation du produit ou une action de sélection d'une autre directive. Ces deux types d'action sont modélisés comme des types spécialisés du concept action, *action d'ingénierie* et *action de délégation* (Figure 3.17).

L'action d'ingénierie peut être de deux types, action atomique et action complexe.

L'action *atomique* consiste à modifier directement le produit en cours de développement.

L'action *complexe* est composée d'autres actions d'ingénierie.

L'action de *délégation* consiste à déléguer la réalisation d'une intention à une autre directive.

Les actions sont exécutées par un outil, une procédure ou un agent humain.

6.3.2.4. Hiérarchie de directives tactiques

Les directives plan, choix et exécutable sont définies récursivement au moyen de la notion de directive tactique. Ce sont des agrégats composés d'autres directives sous forme de hiérarchies de directives. Les feuilles d'une telle hiérarchie sont des directives exécutables ou informelles. Par exemple, la directive ayant la signature « $\langle (Annotation \text{ 'construite'}), \text{Annoter le service} \rangle$ » a le modèle de processus défini sous forme d'une hiérarchie de directives

comportant des directives de type choix et plan et dont les feuilles sont des directives informelles comme nous montrons dans la suite de ce chapitre.

6.3.3. Directive informelle

Une *directive informelle* est une directive qui n'est pas structurée. Elle ne peut pas être décomposée en sous-directives. Elle explique de manière textuelle comment procéder pour obtenir le produit cible. On lui associe une description textuelle.

Une directive informelle ne propose pas de démarche détaillée à suivre pour aboutir au résultat attendu. Elle définit seulement des hypothèses, des règles, des conditions à respecter et des contraintes à ne pas violer. Elle peut aussi définir quel est le type du résultat à obtenir sans préciser pour autant comment procéder formellement pour l'obtenir. Des conseils et des exemples peuvent être proposés à l'utilisateur pour le guider dans la satisfaction de l'intention.

6.4. Directives associées à la Carte

Comme le montre le modèle de Carte de la Figure 3.13, il existe trois types de directives qui sont associées à des éléments de la Carte :

- Directive de réalisation d'intention (DRI) associée à une section ;
- Directive de sélection de stratégie (DSS) associée à un couple d'intention ;
- Directive de sélection d'intention (DSI) associée à une intention.

Une directive (*DXX*) est décrite sous la forme présentée à la Figure 3.19.

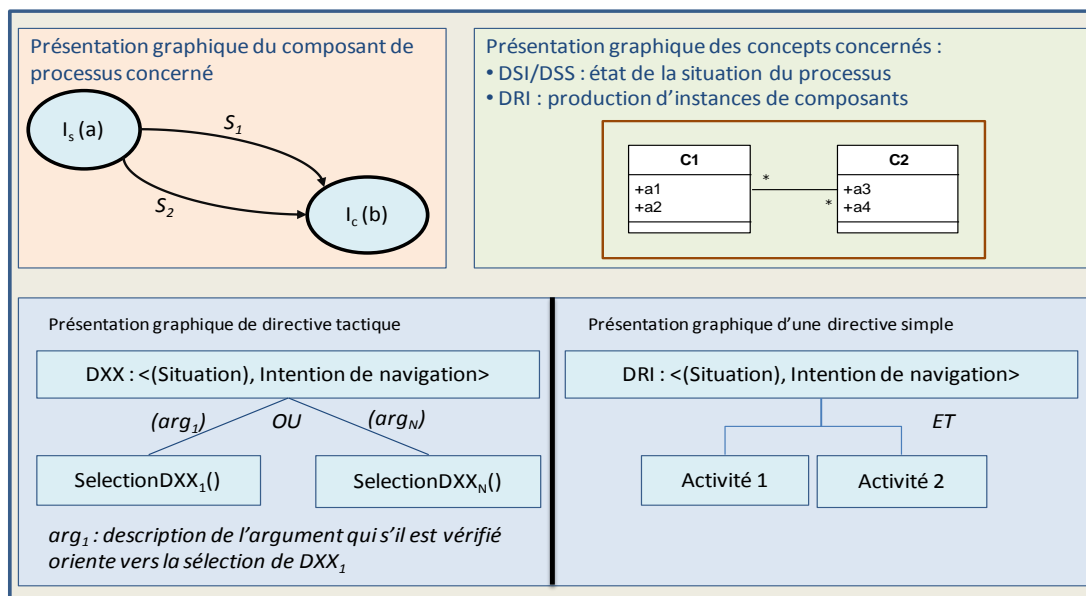


Figure 3.19. Patron de conception d'une directive

6.4.1. Directive de réalisation d'intention

Une *Directive de Réalisation d'Intention (DRI)* aide à la réalisation d'une intention selon une stratégie donnée. Pour chaque section de la carte il existe une DRI fournissant le moyen opérationnel de satisfaire l'intention cible de la section.

Par souci d'homogénéisation, la signature d'une DRI associée à une section $\langle I_i, I_j, S_{ij} \rangle$ est un couple $\langle (\text{situation}), \text{intention} \rangle$ fait comme suit :

La *situation* comporte la partie de produit qui est la cible de l'intention I_i et dont l'état peut être précisé par une condition d'occurrence ;

L'*intention* est exprimée selon la structure proposée par le modèle de but. Elle concerne l'intention cible I_j de la section et inclut la stratégie de la section sous la forme I_j **avec** S_{ij} .

Prenons comme exemple la DRI associée à la section $\langle \text{Annoter les services}, \text{Annoter les services}, \text{Par annotation du verbe} \rangle$. La signature de cette DRI est la suivante :

$\langle (\text{Annotation 'construite'}), (\text{Annoter})_{\text{verbe}} (\text{le service})_{\text{cible}} (\text{par annotation du verbe})_{\text{manière}} \rangle$

La situation de cette signature comporte la partie de produit « *(Annotation 'construite')* ». L'intention de la signature est composée de l'intention « *Annoter le service* » où « *le service* » est la cible de la section et de la stratégie « *par annotation du verbe* » qui est exprimée à l'aide du paramètre *manière*.

Toute DRI est une directive de l'un des trois types possibles, c'est-à-dire informelle, tactique ou stratégique.

6.4.2. Directive de sélection de stratégie

Une *Directive de Sélection de Stratégie (DSS)* détermine quelles sont les stratégies connectant deux intentions et aide à choisir l'une d'entre elles. Elle peut être appliquée lorsque l'intention source et l'intention cible sont déterminées et qu'il y a plusieurs stratégies possibles pour satisfaire l'intention cible à partir de l'intention source. Le rôle de la DSS est de guider la sélection de la stratégie la mieux appropriée pour la situation donnée.

Autrement dit, pour un couple d'intentions connectées par plus d'une stratégie de même direction, il existe une DSS. Comme toute directive, la DSS est définie par un couple $\langle (\text{situation}), \text{intention} \rangle$

Etant donnée la spécificité de cette directive, nous spécialisons sa signature. La signature d'une DSS associée à un couple d'intentions $\langle I_i, I_j \rangle$ est exprimée de la manière suivante :

La situation comporte la partie de produit qui est la cible de l'intention I_i ; une condition d'occurrence peut préciser l'état de cette partie de produit ;

L'intention est exprimée sous la forme : **Progresser** _{verbe} (**vers** I_j) _{cible}.

Le verbe *Progresser* est toujours utilisé pour exprimer les intentions des DSS d'une manière uniforme. De plus, le mot *vers* précise quelle est l'intention cible de la progression.

Par exemple, considérons les deux sections $\langle \text{Démarrer}, \text{Formuler un requête}, \text{Par langage naturel} \rangle$ et $\langle \text{Démarrer}, \text{Formuler un requête}, \text{Par template} \rangle$. Ces deux sections possèdent la même intention source « *Démarrer* » et la même intention cible « *Construire Formuler un requête* ». Une DSS est associée à ce couple d'intentions. La signature de cette DSS est la suivante : « **Progresser** _{verbe} (**vers** *Formuler un requête*) _{cible} ».

Une DSS est toujours une directive tactique. Dans notre exemple, la directive de sélection de stratégie illustrée à la Figure 3.20 est une directive de type choix. Elle aide l'utilisateur à faire son choix parmi les deux sections et à progresser ainsi dans la carte. Les deux alternatives dans cette DSS sont des directives exécutables dont les actions sont de type délégation. Elles délèguent le processus à d'autres directives, dans notre cas, aux DRI associées aux deux sections possibles. En réalité, l'utilisateur sélectionne une DRI associée à la section grâce à l'action de délégation de la directive exécutable et il l'exécute.

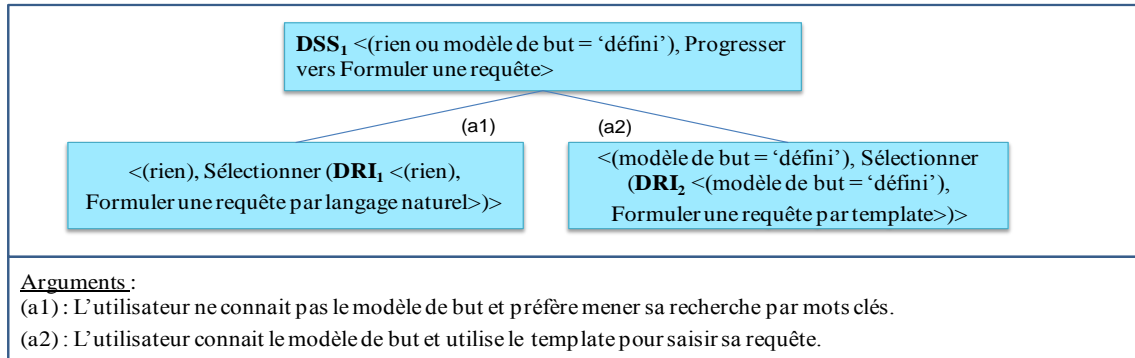


Figure 3.20. Exemple d'une DSS

6.4.3. Directive de sélection d'intention

Une *Directive de Sélection d'Intention (DSI)* détermine quelles sont les intentions qui peuvent succéder à l'intention donnée et aide à choisir l'une d'entre elles. Elle peut être appliquée lorsqu'une intention vient d'être réalisée et que l'utilisateur doit déterminer quelle sera l'intention à réaliser à la prochaine étape. Puisque plusieurs intentions peuvent être accomplies dans la prochaine étape, le rôle de la DSI est de guider la sélection de l'intention suivante et de fournir l'ensemble des DRI et DSS correspondantes.

La signature d'une DSI associée à une intention I_i de la carte est exprimée de la manière suivante :

La situation comporte la partie de produit qui est la cible de l'intention I_i ; elle peut être précisée par une condition d'occurrence ;

L'intention est exprimée sous la forme : **Progresser** *verbe* (**depuis** I_i) *source*

L'intention de la signature de la DSI exprime le fait que la directive aide l'utilisateur à progresser dans la carte. Le verbe *Progresser* est utilisé à ce propos. Le paramètre *source* exprime l'intention qui est la source de progression. De plus, le mot *depuis* précise quelle est l'intention source de la progression.

La Figure 3.21 montre un exemple d'une DSI. La partie supérieure de la figure indique que pour progresser à partir de l'intention « *Formuler une requête* », on a trois possibilités : aller vers l'intention « *Proposer des services* », aller vers l'intention « *Formuler une requête* » ou terminer le processus par la sélection de l'intention « *Arrêter* ». Une DSI est associée à l'intention « *Formuler une requête* ». La signature de cette DSI est « **Progresser** *verbe* (**depuis** *Formuler une requête*) *source* ».

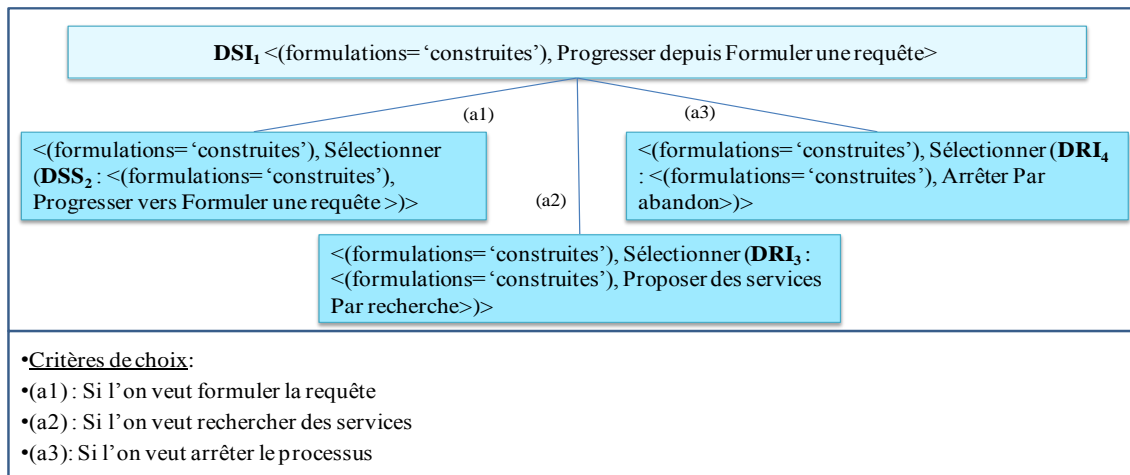


Figure 3.21. Exemple d'une DSI

7. CONCLUSION

Nous avons introduit dans ce chapitre l'approche *PASIS* qui permet de traiter le problème de la publication et de la recherche de services intentionnels. Ainsi que les caractéristiques de l'approche qui sont : l'expression orientée métier, le processus de guidage méthodologique, l'utilisation des ontologies et l'usage de l'algorithme d'appariement.

Dans cette approche, nous avons introduit une couche conceptuelle entre la couche métier et la couche logique. Cette couche est ajoutée pour surmonter le problème d'écart entre le niveau métier et le niveau logique à la fois au niveau de la publication et au niveau de la recherche de services et éventuellement, éviter la modification de la structure de l'annuaire. Ce niveau conceptuel constitue l'essentiel de l'apport de l'approche *PASIS* grâce aux modèles (MiS et but) et aux processus de guidage permettant de faire le lien entre le niveau métier et l'annuaire étendu. Ainsi, l'approche proposée n'a pas modifié ni la structure de l'annuaire, ni les algorithmes utilisés pour la recherche.

Nous avons décrit les trois ontologies utilisées dans *PASIS* : *iSOnto* (ontologie technique du modèle de but), *vOnto* (ontologie générique des verbes) et *pOnto* (ontologie de domaine des produits). Les processus de publication et de recherche de services utilisent très largement ces ontologies pour capturer, du côté fournisseur, la sémantique du descripteur de service intentionnel et du côté utilisateur, le besoin à satisfaire.

Nous avons montré comment l'annotation est utilisée comme une technique d'extension sémantique vers les trois ontologies : *iSOnto*, *vOnto* et *pOnto* pour la spécification et l'implémentation du descripteur intentionnel de service.

Enfin, le modèle de Carte et ses directives méthodologiques est présentée. Ce modèle est utilisé pour formaliser les processus de guidage de l'approche *PASIS*.

CHAPITRE 4

PUBLICATION DE SERVICES INTENTIONNELS

1. INTRODUCTION

Nous introduisons dans ce chapitre, le processus de guidage de publication que le fournisseur utilise pour annoter les services. Ce processus est décrit en utilisant le modèle MAP. Un Map est alors considéré comme une structure de navigation contenant un nombre fini de chemins, aucun n'étant recommandé « à priori », mais chacun étant choisi de manière dynamique. La sélection d'une stratégie se fait au fur et à mesure de la réalisation des intentions, en fonction de l'état du produit. Le fournisseur sélectionne une intention pour progresser dans le processus et réalise l'intention sélectionnée à l'aide des directives associées au Map. Ce chapitre introduit l'ensemble des directives qui guident le processus de la publication.

Le reste du chapitre est organisé de la façon suivante : la section 2 présente l'étape d'identification des services intentionnels en utilisant la carte et le modèle intentionnelle de services. Ensuite, la section 3 présente le processus « Map-Publication » pour la publication de services en présentant le sommaire des différentes directives associées. Enfin, la section 4 correspond à la conclusion du chapitre.

2. ÉTAPE PREALABLE D'IDENTIFICATION DES SERVICES

Comme le montre la Figure 4.1, le processus de publication nécessite une étape préalable qui est l'identification des services intentionnels à partir d'une modélisation intentionnelle des processus métier. Ces services intentionnels sont décrits selon le modèle MiS qui adopte le système de représentation de la *Carte* [Rolland, 2000] pour découvrir les services intentionnels. Cette démarche donne à l'agent métier les moyens de construire un modèle du

business à partir duquel il est aisé d'identifier les services, de les décrire selon les termes du modèle intentionnel de services MiS [Kaabi, 2007].

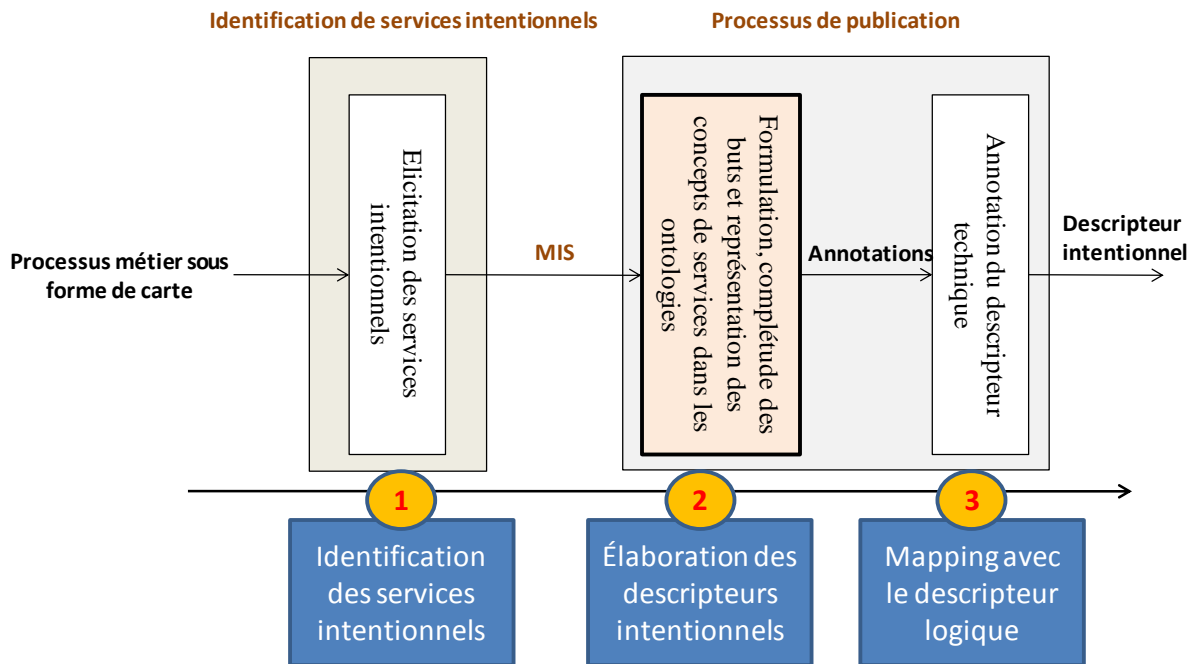


Figure 4.1. *Étapes de la publication*

L'étape suivante consiste à utiliser le processus de publication pour aider le fournisseur dans sa démarche d'annotation des services intentionnels identifiés. Finalement, la dernière étape consiste à implémenter le descripteur de service dans un annuaire étendu afin de favoriser la recherche dirigée par les buts.

Dans le reste de cette section, nous expliquons la démarche d'identification des services intentionnels à partir d'une carte et à l'aide du modèle MiS.

2.1. Utilisation de la Carte

La *Carte* est un système de représentation qui permet de modéliser les processus dans des termes intentionnels [Rolland et al., 1999] [Rolland, 2007]. Cette représentation explicite de la variabilité offerte par les cartes est manquante dans d'autres formalismes tels que TROPOS [Bresciani et al., 2004] ou KAOS [Lamsweerde et al., 2000]. Elle offre un mécanisme de représentation basé sur un ordonnancement non déterministe d'*intentions* et de *stratégies*. Le graphe montre quelles intentions peuvent être réalisées au moyen de quelles stratégies une fois qu'une intention a elle-même été préalablement réalisée. Les cartes proposent aussi un mécanisme d'affinement. L'affinement permet de décrire les intentions et stratégies à différents niveaux de détail [Rolland, 2007].

La carte est adaptée à la découverte des services. En effet, une carte met en évidence les différentes stratégies pour satisfaire le même but et aussi les différentes combinaisons de stratégies et de buts pour atteindre un but. Elle aide aussi à la découverte de la variabilité des services : chaque combinaison possible de buts et de stratégies identifie une variante possible de services pour atteindre le but du service global.

Finalement, on peut identifier les services (selon le modèle de MiS) de manière systématique à partir de la carte et les représenter. Ces services seront annotés avec les descriptions sémantiquement pour alimenter l'annuaire des services.

2.2. La découverte des services intentionnels

La Figure 4.2 montre un exemple de carte *Satisfaire efficacement les besoins en produits* [kaabi, 2007]. La carte comporte quatre buts (*Démarrer*, *Acquérir des produits*, *Contrôler le stock* et *Arrêter*), douze stratégies (*Par seuil de réapprovisionnement*, *Par prévisions stratégiques*, *Manuellement*, *Entrer en stock des produits livrés*, *Entrer les produits directement*, *Périodique*, *Echantillonnage*, *En continu*, *Par évaluation*, *Par inspection de la qualité*, *Par transfert de produits*, *Par contrôle du paiement des factures*) et douze sections.

Chaque but est codé par une lettre de l'alphabet. A la Figure 4.2 par exemple, les quatre buts de la carte à savoir *Démarrer*, *Acquérir des produits*, *Contrôler le stock* et *Arrêter* sont respectivement codés par a, b, c et d.

Les stratégies sont numérotées relativement à leurs buts cible et source. Ainsi, deux stratégies ayant les mêmes buts source et cible seront respectivement numérotées par deux numéros distincts. Une carte peut donc avoir plusieurs stratégies numérotées de la même manière. Par exemple, à la Figure 4.2, les stratégies de la carte sont codées comme suit : 1, 2 et 3 (ab) ; 1 (bc) ; 1, 2, 3, 4, 5 et 6 (cc) ; 1 (cd) ; 1(ac).

Les sections sont codées par juxtaposition de trois éléments (i) la lettre du but source, (ii) la lettre du but cible et (iii) le numéro de la stratégie. Par exemple, la section ab1 permet, en partant du but a, d'atteindre le but b en suivant la stratégie 1. La Figure 4.2 représente un exemple de codification locale d'une carte. Nous avons les douze sections suivantes : ab₁, ab₂, ab₃, bc₁, cc₁, cc₂, cc₃, cc₄, cc₅, cc₆, cd₁ et ac₁.

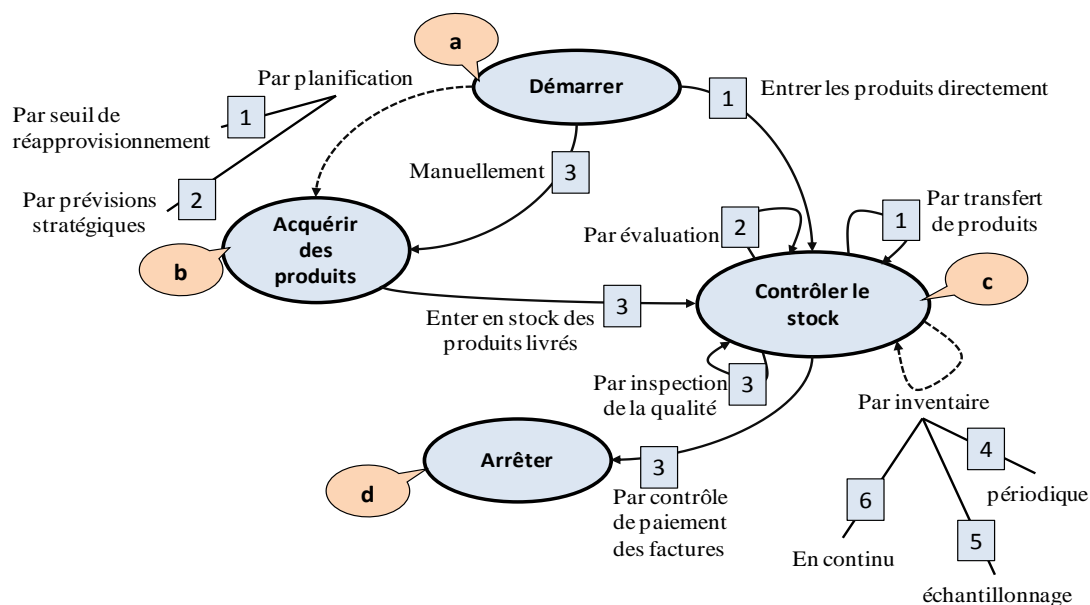


Figure 4.2. Exemple d'une carte *Satisfaire efficacement les besoins en produits*

Une carte est composée d'une ou plusieurs sections. Une section est un triplet <une intention source I, une intention cible J, une stratégie S> qui capte la manière spécifique pour atteindre

l'intention cible J à partir de l'intention source I avec la stratégie S. Par exemple, < *Acquérir des produits*, *Contrôler le stock*, *Entrer en stock des produits livrés* > représente un moyen de parvenir à l'intention cible *Contrôler le stock* de l'intention source *Acquérir des produits* en suivant la stratégie *Entrer en stock des produits livrés*.

2.2.1. Le modèle intentionnel de services

Le concept de service intentionnel qui abstrait les détails du service logiciel et de ses fonctionnalités pour se concentrer sur son essence, à savoir le but qu'il permet d'atteindre [Kaabi, 2007]. Les services intentionnels sont spécifiés par le modèle intentionnel de services (MiS). La Figure 4.3 présente le méta-modèle MiS en utilisant les notations du diagramme de classes UML et elle montre, par le jeu des trois couleurs utilisées que la description d'un service intentionnel comporte trois parties correspondant à son *interface*, l'expression de son *comportement* et celle de sa *composition* [Kaabi, 2007] [Rolland et al., 2008].

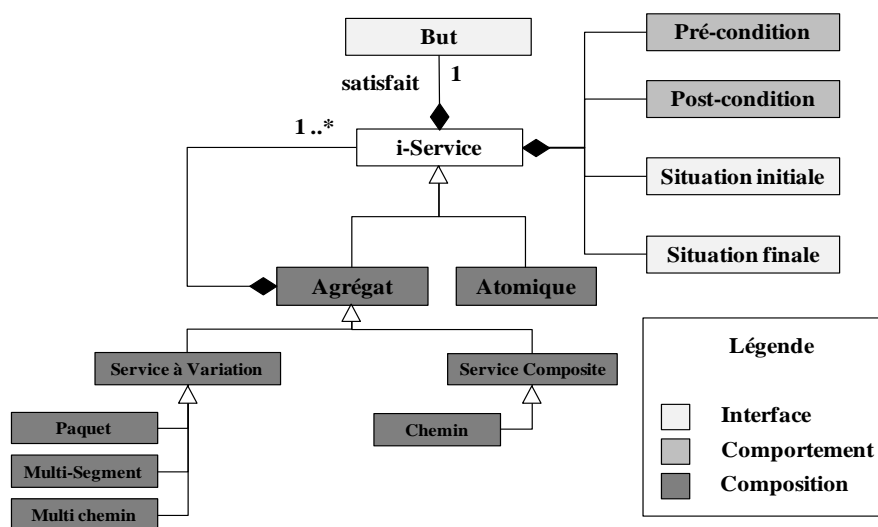


Figure 4.3. Le modèle du service intentionnel

L'interface. Il ya trois éléments qui constituent l'interface : le but, la situation initiale et la situation finale. L'idée clé de MIS est qu'un service intentionnel permet la réalisation d'une intention donnée à partir d'une situation initiale et en se terminant avec une situation finale.

Le comportement. La pré-condition et la post-condition sont respectivement l'état initial et l'état final, c-à-dire, l'état requérant la réalisation de l'intention et l'état résultant de sa réalisation.

La composition. Le service atomique et le service *agrégat* sont impliqués dans l'aspect de composition de services. Un service atomique a une intention opérationnalisable qui peut être atteinte directement par un service atomique opérationnel. Un service *agrégat* qui satisfait une intention de haut niveau, doit être décomposé jusqu' à ce que des services atomiques opérationnels soient identifiés.

La Figure 1.5 met en évidence que le concept central du modèle est celui de *service intentionnel*. On observe aussi que la perspective intentionnelle choisie dans cette thèse se traduit par le fait qu'un service exhibe une intentionnalité formulée par le but qu'il permet à ses utilisateurs d'atteindre. Sachant qu'un *but* reflète une *intention*, un objectif que l'on

cherche à atteindre sans pour autant dire comment l'atteindre (par exemple : *Payer une réservation*).

2.2.2.L'identification des services intentionnels à partir de la carte

Une carte métier présente trois types de relations qui relient les sections entre elles : (i) la relation *paquet*, (ii) la relation *Multi-segment*, (iii) et la relation *Chemin*. En général, une carte est considérée comme un *multi-chemin* depuis son intention *démarrer* vers son intention *arrêter*, et peut contenir des *multi-segments*.

La relation paquet : On a une relation de type paquet entre plusieurs sections quand ces sections ont le même couple d'intentions (source et cible) et que le choix d'une de ces sections pour la réalisation de l'intention cible empêche la sélection des autres sections. Il s'agit de plusieurs sections mutuellement exclusives.

La relation Multi-segment : Dans une carte, il est possible de réaliser une intention cible à partir d'une intention source en utilisant plusieurs stratégies complémentaires. Chacune de ces stratégies, couplée avec l'intention source et l'intention cible, définit une section dans la carte. Cette topologie est appelée multi-segment (multithread).

La relation Chemin : cela établit un rapport de précédent/succession entre les sections. Par exemple, les sections *< Démarrer, Contrôler le stock, Entrer les produits directement >* et *< Contrôler le stock, Arrêter, Par contrôle du paiement des factures >* constituent un chemin.

La relation multi-chemin : il est possible d'atteindre une même intention cible, à partir d'une intention source, en suivant plusieurs chemins de la carte.

Afin d'obtenir des services intentionnels et leur composition à partir d'une carte métier, nous suivons trois orientations stratégiques [Rolland et al., 2007] :

- ✓ La première ligne directrice consiste à associer toutes les sections non-raffinées d'une carte à un service atomique. Dans le cas de la carte de l'application *Satisfaire efficacement les besoins en produits*, cette correspondance conduit à un ensemble de 12 services atomiques, y compris par exemple: $ab_3 \rightarrow S_{\text{Accueillir des produits manuellement}}$ et $cc_3 \rightarrow S_{\text{contrôler le stock par inspection de la qualité}}$.
- ✓ La deuxième orientation consiste à identifier les chemins d'une carte en appliquant un algorithme qui calcule les chemins dans un graphe. Cet algorithme est une adaptation de l'algorithme de [MacNaughton et Yamada, 1960].
- ✓ La troisième orientation consiste à déterminer les services agrégats en établissant les correspondances suivantes entre les relations des sections dans la carte et les types de service agrégats dans le modèle MiS : *<Chemin-Composite>*, *<Paquet-Alternative>*, *<Multi-segment-Choix>*, et *<Multi-chemin-Multi-chemin>*. Dans le cas de la carte de l'application *Satisfaire efficacement les besoins en produits*, nous avons identifié un ensemble de 7 services agrégats.

Dans la figure 4.2, il existe un paquet entre les deux buts *début* et *Acquérir des produits*. Le nom de ce paquet est *Par planification*. Il regroupe deux moyens alternatifs pour accueillir des produits : *Par seuil de réapprovisionnement* ou *Par prévisions stratégiques*. L'utilisation de l'un de ces deux moyens exclut l'utilisation de l'autre.

Il existe aussi deux manières différentes pour *Acquérir des produits*, *Manuellement* ou *par planification*. Les deux stratégies *Par Manuellement* et *Par planification* ont le même but source *Démarrer* et le même but cible *Acquérir des produits*. Les deux sections *< Démarrer,*

Acquérir des produits, Manuellement > et < *Démarrer, Acquérir des produits, Par planification* > forment une topologie multi-segment.

Finalement, il existe deux façons distinctes d'atteindre l'intention *Contrôler le stock* à partir de *Démarrer*. La première est le chemin par le biais de *Démarrer, Acquérir des produits, Contrôler le stock* et *Arrêter*. La seconde est le chemin *Démarrer, Contrôler le stock* et *Arrêter*.

3. PROCESSUS DE PUBLICATION

Ce processus est modélisé en utilisant le formalisme de la Carte. La Map-Publication présentée à la Figure 4.4 définit. Outre les intentions *Démarrer* et *Arrêter* que l'on retrouve dans toute carte, la principale intention du processus, est : *Annoter le service*.

L'intention *Annoter le service* recouvre les composantes du processus qui concernent l'annotation des services.

L'intention *Arrêter* recouvre les composantes du processus qui concernent la phase terminale de mise en place de l'annotation en vérifiant la complétude des différents éléments de l'annotation.

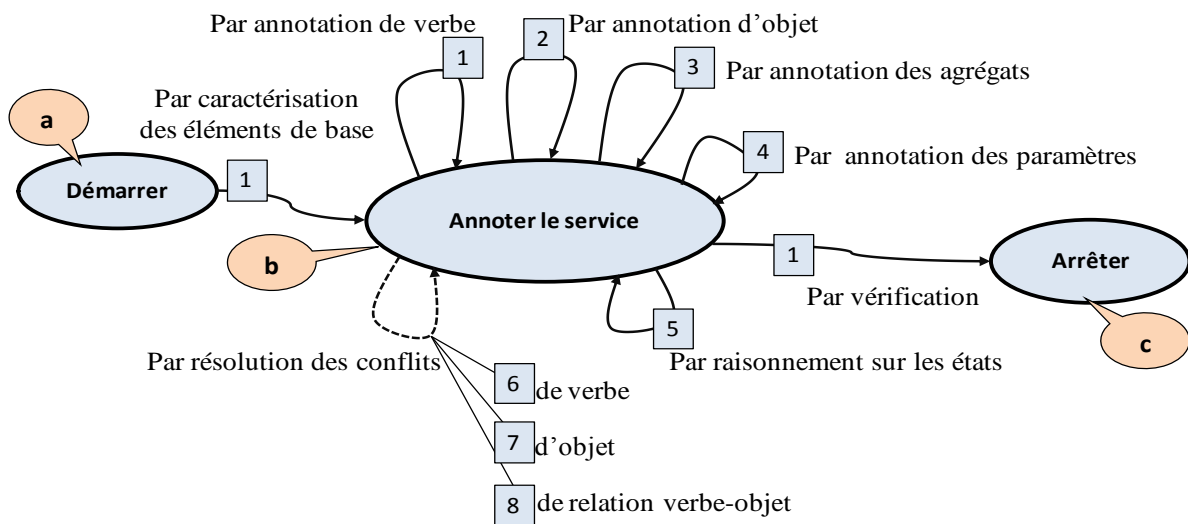


Figure 4.4. Map-Publication

Les 9 sections qui composent la MAP-Publication, proposent des stratégies pour la réalisation de ces deux intentions.

- La stratégie *Par caractérisation des éléments de base* a été introduite dans la perspective de capturer les éléments constituant les annotations en utilisant le modèle de but et le modèle MiS. L'entrée de cette stratégie sont les éléments non structurés de l'annotation sémantique. La sortie de cette stratégie sont les éléments constituant l'annotation du service intentionnel structurés selon l'ontologie de services intentionnels *iSOnto*. Dans cette structuration, les éléments constituant le but sont identifiés, ainsi que les éléments constituant les composants du service intentionnel.
- La stratégie *Par annotation du verbe* a été introduite dans la perspective d'identifier le verbe dans le but, préciser sa classe (de sens de verbe) et de l'annoter.

- La stratégie **Par annotation de l'objet** a été introduite dans la perspective d'identifier l'objet dans le but, de vérifier son appartenance à l'ontologie des produits de domaine et de l'annoter.
- La stratégie **Par annotation des agrégats** a été introduite dans la perspective d'ajouter les composants d'un service agrégat.
- La stratégie **Par raisonnement sur les états** a été introduite dans la perspective d'ajouter les états du service selon le type de la classe du sens de verbe.
- La stratégie **Par résolution des conflits** a été introduite dans la perspective de résoudre les conflits du verbe, de l'objet ou bien la relation entre verbe et objet. Cette stratégie consiste à vérifier la compatibilité du verbe et du produit en utilisant l'ontologie des produits.
- La stratégie **Par vérification** a été introduite dans la perspective de vérifier la complétude des annotations.

3.1. Sommaire des directives du Map-Publication

Conformément au modèle MAP, chaque directive de type DSI a une signature composée d'une situation et d'une intention. De plus, un identifiant unique est systématiquement employé pour rapidement désigner les DSI. L'identifiant est de la forme DSI_n , où n est un numéro d'intention. La directive DSI_n est définie pour guider la sélection de la prochaine intention à réaliser sachant que l'intention I_n a déjà été elle-même réalisée.

La Map-Publication (Figure 4.4) a trois intentions : « Démarrer », « Annoter le service » et « Arrêter ». Elle propose une stratégie pour progresser de l'intention source « Démarrer » vers l'intention cible « Annoter le service ». Mais, pour progresser de « Annoter le service » vers « Annoter le service », il propose 6 stratégies et une stratégie de « Annoter le service » vers « Arrêter ». Par conséquent, comme le montre le Tableau 4.1, Map-Publication a une DSS. Cette DSS est référencée par un identifiant unique de type DSS_n où n est un numéro séquentiel unique pour les DSS d'une Map. Une notation pointée est employée pour garantir l'unicité globale des numéros de DSS.

La Map-Publication contient sept sections. A chacune de ces sections une DRI est associée. Les sept DRI correspondant aux sections du Map-Publication sont identifiées au Tableau 4.1. Tout comme les DSS, les DRI sont référencées au moyen d'un identifiant de type DRI_n où n est unique relativement au Map concerné. Afin de garantir une unicité globale, la notation pointée est employée pour tous les Map autres que Map-publication. Ainsi, les DRI du Map M_i sont numérotées $DRI_{i,1}$, $DRI_{i,2}$, $DRI_{i,3}$, etc.

Le Tableau 4.1 comprend deux types de lignes faisant correspondre, la DSS et les DRI référencées par cette DSS. Ainsi, trois colonnes faisant correspondre, pour chaque directive, son identifiant, son type et la sous-section qui la détaille dans ce chapitre.

Tableau 4.1. Les directives du Map-Publication

Les sections de la carte Map-Publication	Type	Section
ab_1 : <Démarrer, Annoter le service, Par caractérisation des éléments de base>	DRI_1	3.2.1
Progresser vers Annoter le service	DSS_1	3.2.2
<ul style="list-style-type: none"> bb_1 : < Annoter le service, Annoter le service, Par annotation du verbe> 	DRI_2	3.2.2.1

• bb_2 : < Annoter le service, Annoter le service, Par annotation d'objet>	DRI ₃	3.2.2.2
• bb_3 : < Annoter le service, Annoter le service, Par annotation des agrégats>	DRI ₄	3.2.2.3
• bb_4 : < Annoter le service, Annoter le service, Par annotation des paramètres>	DRI ₅	3.2.2.4
• bb_5 : < Annoter le service, Annoter le service, Par raisonnement sur les états>	DRI ₆	3.2.2.5
• bb_6 : < Annoter le service, Annoter le service, Par résolution des conflits >	DRI ₇	3.2.2.6
bc_1 : <Annoter le service, Arrêter, par vérification>	DRI ₈	3.2.3

3.2. Exploration du MAP-Publication

Dans cette section, nous présentons les directives DSS et DRI identifiées à la section 3.1. La présentation des directives est regroupée suivant la progression dans le Map. Trois principales sous sections sont ainsi décrites suivant la progression depuis « Démarrer » et « Annoter le service ».

3.2.1. Réaliser « Annoter le service Par caractérisation des éléments de base » (DRI₁)

La directive DRI₁ est une directive associée à la section bb_2 : <Démarrer, Annoter le service, Par caractérisation des éléments de base>. Elle permet de progresser à partir de l'intention « Démarrer ». Le Map-Publication indique qu'à partir de la situation initiale il y a un service à décrire, il est possible de progresser en sélectionnant la seule intention « Annoter le service ». Par conséquent, il n'y a qu'une seule directive de type DRI, c'est la directive de réalisation d'intention DRI₁ (Figure 4.5).

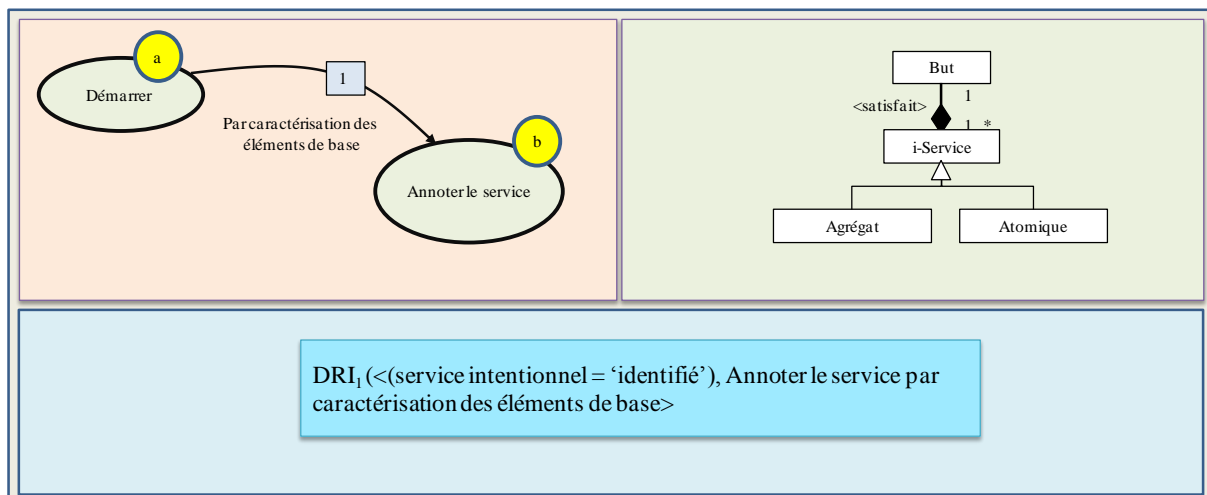


Figure 4.5. La structure de la DRI₁

La Directive DRI₁ (< (service intentionnel = ‘identifié’), Annoter le service Par caractérisation des éléments de base >) guide la stratégie de réalisation de l’intention « *Annoter le service* », (Figure 4.5-partie supérieure). Le processus de la publication commence par cette directive après l’identification des services intentionnels à partir de la carte.

En effet, le fournisseur souhaite effectuer une annotation d’un service, il connaît l’ontologie *iSonto* et il structure les éléments de l’annotation selon cette ontologie. Tout d’abord, il saisit les éléments constituant le but ou une partie de but exprimé dans l’intention du service. Ensuite, il définit le type de service (atomique ou agrégat). L’entrée de cette directive est le service intentionnel identifié. La sortie de cette directive est l’annotation du service intentionnel structurée selon l’ontologie de services intentionnels *iSonto*.

Prenons comme exemple le service atomique *S Contrôler le stock manuellement* de *Satisfaire efficacement les besoins en produits* (Figure 4.2), le but du service est « *Contrôler le stock manuellement* » et nous ajoutons que le type du service est « *atomique* ». Finalement, nous obtenons les éléments de base pour l’annotation :

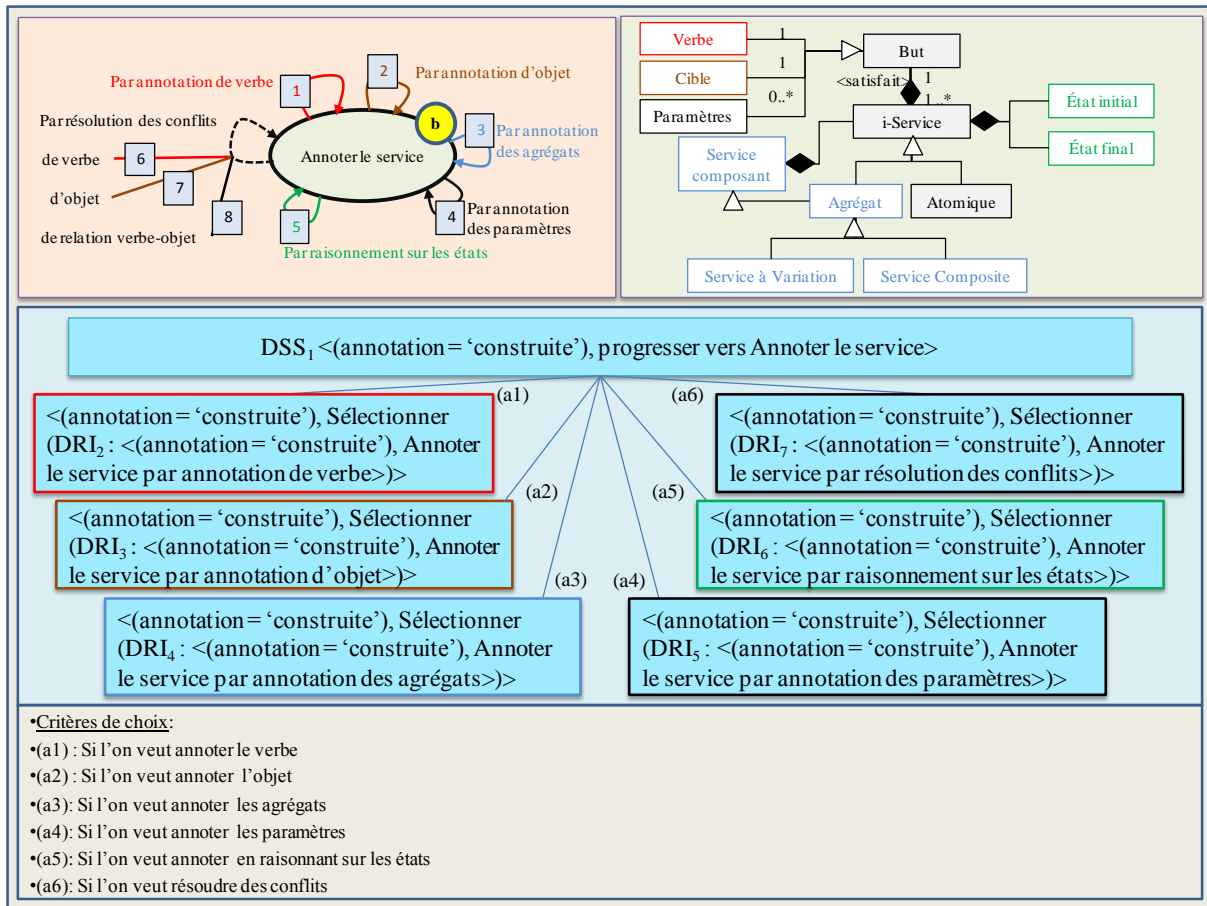
(*Contrôler le stock manuellement*) but (*S Contrôler le stock manuellement*) atomique

3.2.2. Progresser vers « *Annoter le service* » (DSS₁)

La DSS₁ est de type choix. Elle permet de progresser à partir de l’intention *Annoter le service* vers l’intention *Annoter le service*, c’est-à-dire qu’elle permet de boucler sur l’intention elle-même. Elle offre six stratégies pour réaliser cela : *Par annotation du verbe*, *Par annotation de l’objet*, *Par annotation des agrégats*, *Par annotation des paramètres*, *Par raisonnement sur les états* et *Par résolution des conflits (de verbe, de l’objet, de verbe-objet)*. La Figure 4.6 montre la structure de cette DSS₁ ainsi que les arguments pour le choix d’une stratégie plutôt qu’une autre.

Dans la suite, nous détaillons les 6 DRI associées à DSS₁ qui ont été identifiées. Ces DRI sont :

- DRI₂ : < (annotation = ‘construite’), Annoter le service par annotation du verbe>
- DRI₃ : < (annotation = ‘construite’), Annoter le service par annotation de l’objet>
- DRI₄ : < (annotation = ‘construite’), Annoter le service par annotation des agrégats>
- DRI₅ : < (annotation = ‘construite’), Annoter le service par annotation des paramètres>
- DRI₆ : < (annotation = ‘construite’), Annoter le service par raisonnement sur les états>
- DRI₇ : < (annotation = ‘construite’), Annoter le service par résolution des conflits>

Figure 4.6. La structure de DSS₁

3.2.2.1. Réaliser « Annoter le service par annotation du verbe » (DRI₂)

La DRI₂ est associée à la section < Annoter le service, Annoter le service, par annotation du verbe> de la carte Map-Publication (Figure 4.4). Elle vise à réaliser l'intention *Annoter le service* en utilisant la stratégie *par annotation du verbe*.

Ceci consiste d'abord, à coupler les éléments du but de service avec les éléments du formalisme de but. Le couplage est réalisé en identifiant la nature de chaque élément de but et en le reliant à l'élément correspondant du formalisme.

Prenons comme exemple le service atomique *S* *Contrôler le stock manuellement* de *Satisfaire efficacement les besoins en produits* (Figure 4.2), le but du service est « *Contrôler le stock* », ce but a le verbe « *Contrôler* » :

((*Contrôler*) verbe (*le stock manuellement*)) but (*S* *Contrôler le stock manuellement*) atomique

L'identification du verbe dans le but nous aide à préciser sa classe de sens. Une classe de sens de verbe est une sous-classe des seize classes rencontrées dans l'ontologie des verbes [Urrego, 2005]. La classe de sens permet de nous assurer du sens du verbe même. Dans cette classe des verbes, nous retrouvons tous les verbes qui ont le sens que notre verbe (on peut remplacer le verbe par un autre plus fréquent appartenant à la même classe de sens de verbe). On utilise cette directive d'abord, pour s'assurer du sens de verbe, parce qu'un verbe appartient normalement à une ou plusieurs classes de sens de verbes. Par conséquent, cette directive aide

l'utilisateur dans la précision de la classe de sens de verbe, en donnant plus d'information sur la classe de sens de verbe et des exemples d'utilisation. Par exemple, le verbe « *contrôler* » appartient à la classe de sens de verbes *de pratique de contrôle* qui contient les verbes « *inspecter* », « *surveiller* », « *observer* »...

3.2.2.2. Réaliser « Annoter le service par annotation d'objet » (DRI₃)

La DRI₃ est associée à la section < *Annoter le service, Annoter le service, par annotation d'objet* > de la carte Map-Publication (Figure 4.4). Elle vise à réaliser l'intention *Annoter le service* en utilisant la stratégie *par annotation d'objet*.

Cette directive consiste à identifier l'objet dans le but et à vérifier son appartenance à l'ontologie des produits. Dans cette directive, on prend le concept d'objet, le localise dans l'ontologie des produits, puis on rend tous les concepts adjacents à cet objet pour donner au fournisseur la possibilité de remplacer l'objet pour d'autres concepts similaires.

Dans l'exemple du service atomique *S Contrôler le stock manuellement* de *Satisfaire efficacement les besoins en produits* (Figure 4.2), le but du service est « *Contrôler le stock* », ce but a l'objet « *stock* » :

(*Contrôler (le stock) objet manuellement*) but (*S Contrôler le stock manuellement*) atomique

3.2.2.3. Réaliser « Annoter le service par annotation des agrégats » (DRI₄)

La DRI₄ est associée à la section < *Annoter le service, Annoter le service, par annotation des agrégats* > de la carte Map-Publication (Figure 4.4). Elle vise à réaliser l'intention *Annoter le service* en utilisant la stratégie *par annotation des agrégats*.

L'annotation des agrégats consiste d'abord à ajouter les composants du service (*type du service agrégat, composants*). La complétude est réalisée conformément à l'ontologie de services intentionnels *iSOnto*,

En appliquant DRI₄ sur le service agrégat *S payer une facture*, on ajout que le type du service est « *aggregate* » ou plus précisément, « *paquet* ». Dans ce cas, nous ajoutons les composants de ce service, qui sont les services atomiques *S payer une facture par chèque*, *S payer une facture par carte* de Crédit, et *S payer une facture par virement*.

(*Payer*) verb (*une facture*) objet (*S payer une facture*) paquet (*S payer une facture par chèque*) service component (*S payer une facture par carte*) service component (*S payer une facture par virement*) service component

3.2.2.4. Réaliser « Annoter le service par annotation des paramètres » (DRI₅)

La DRI₅ est associée à la section < *Annoter le service, Annoter le service, par annotation des paramètres* > de la carte Map-Publication (Figure 4.4). Elle vise à réaliser l'intention *Annoter le service* en utilisant la stratégie *par annotation des paramètres*.

L'annotation des paramètres consiste à identifier les paramètres et les annoter conformément à l'ontologie de services intentionnels *iSOnto*,

Dans l'exemple du service atomique *S Contrôler le stock manuellement* de *Satisfaire efficacement les besoins en produits* (Figure 4.2), le but du service est « *Contrôler le stock* », ce but a un paramètre de type « *manière* » qui est « *manuellement* » :

(*Contrôler le stock (manuellement) manière*) but (*S Contrôler le stock manuellement*) atomique

3.2.2.5. Réaliser « Annoter le service par raisonnement sur les états » (DRI_6)

Les états d'un service sont affectés par la catégorie du sens de verbe, c'est-à-dire que le sens de verbe indique que le paramètre *cible* peut passer d'un état à un autre. Pour cette raison, nous allons utiliser la catégorie du verbe pour raisonner sur les états du service.

La DRI_6 est associée à la section < Annoter le service, Annoter le service, par raisonnement sur les états > de la carte Map-Publication (Figure 4.4). Elle vise à réaliser l'intention *Annoter le service* en utilisant la stratégie *par raisonnement sur les états*. Elle est de type choix et composée de deux sous-directives :

- $DRI_{6,1}$ pour raisonner sur les verbes de type causatif
- $DRI_{6,2}$ pour raisonner sur les verbes de type descriptif

Le type de verbe indique le type de l'intervention *directe* (verbe causatif) ou *indirecte* (verbe descriptif) sur le paramètre *cible*. La Figure 4.7 montre la structure de DRI_6 .

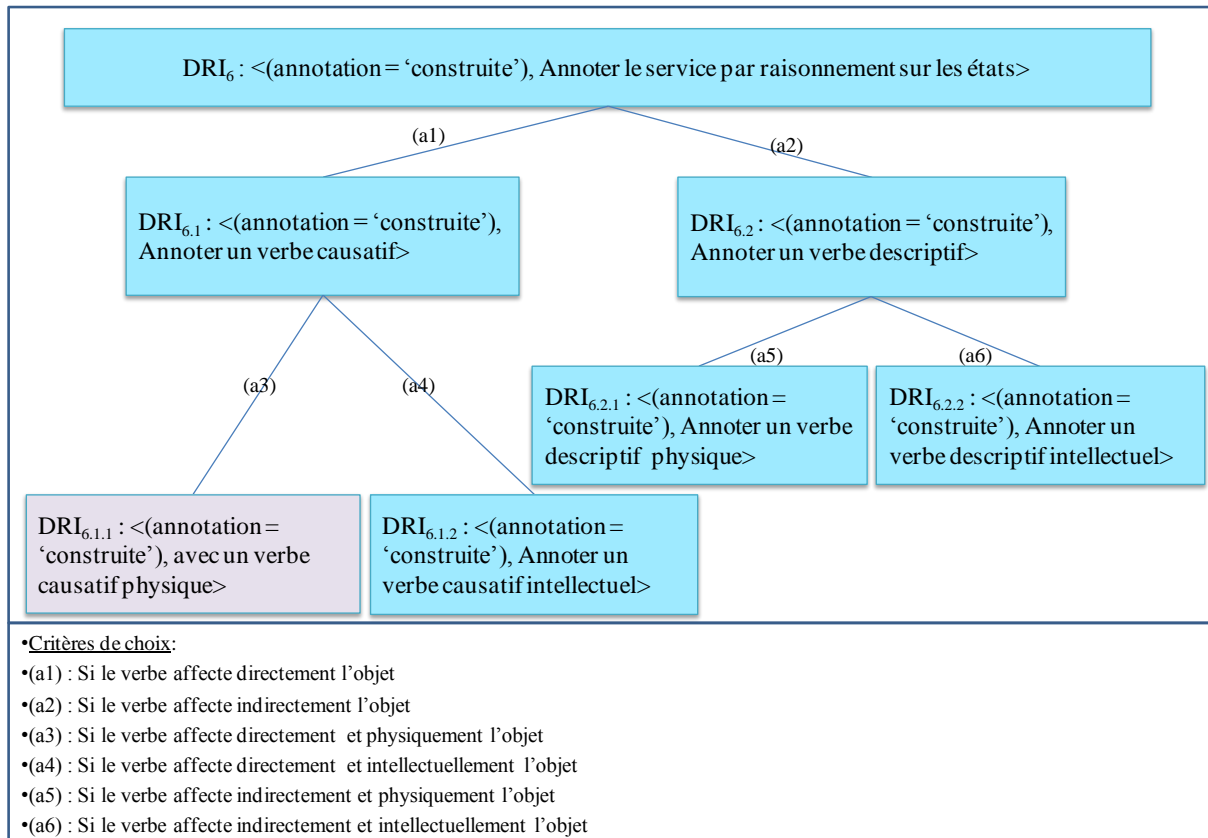


Figure 4.7. Structure de la DRI_6

3.2.2.5.1. Réaliser « raisonner sur les verbes causatifs » ($DRI_{6,1}$)

Une fois le type de verbe identifié comme causatif, la $DRI_{6,1}$ permet de raisonner sur les états. Cette directive est de type choix. Elle offre deux possibilités comme le montre la Figure 4.7 :

- $DRI_{6,1,1}$ pour raisonner sur les verbes de type causatif physique

- DRI_{6.1.2} pour raisonner sur les verbes de type causatif intellectuel

3.2.2.5.1.1. Réaliser « raisonner sur les verbes causatifs physiques » (DRI_{6.1.1})

La DRI_{6.1.1} permet de raisonner sur les états des verbes de type causatif physique. Cette directive est de type choix. Elle offre quatre possibilités comme le montre la Figure 4.8 :

- DRI_{6.1.1.1} pour raisonner sur les verbes de type causatif physique de service
- DRI_{6.1.1.2} pour raisonner sur les verbes de type causatif physique de processus
- DRI_{6.1.1.3} pour raisonner sur les verbes de type causatif physique d'activité
- DRI_{6.1.1.4} pour raisonner sur les verbes de type causatif physique d'action

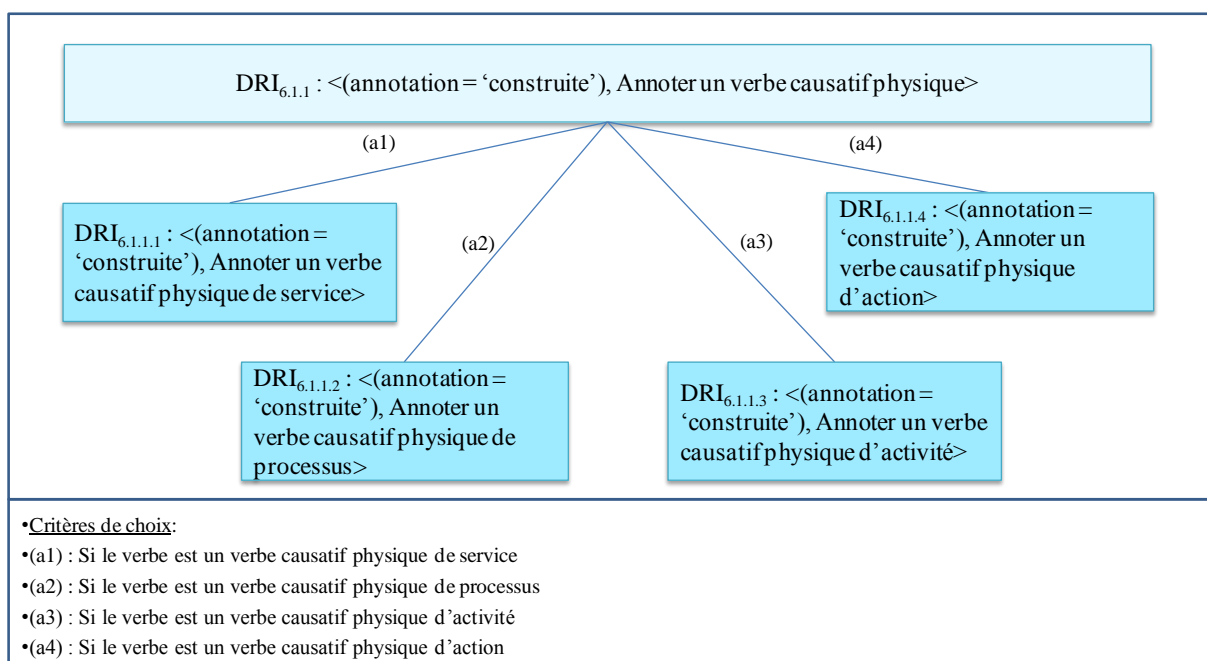


Figure 4.8. Structure de DRI_{6.1.1}

3.2.2.5.1.1.1. Réaliser « raisonner sur les verbes causatifs physiques de service » (DRI_{6.1.1.1})

La DRI_{6.1.1.1} permet de raisonner sur les états des verbes de type causatif physique de service. Un verbe de *service* énonce un développement par étapes. Les verbes de cette catégorie considèrent le changement de *situation*. Autrement dit, les verbes de *service* acceptent les paramètres *situation* (*situation initiale* ou *situation finale* ou les deux). Pour les verbes de *service* les situations décrites par les paramètres *situation initiale* et *situation finale* représentent des *processus*, c'est-à-dire que ces *situations* sont énoncées par des verbes de *processus* et ont aussi un développement par étapes.

Les *situations* énoncées par les verbes de service *causatifs* sont des *situations* de leur paramètre *objet*, ce qui implique le changement d'états de l'*objet*. On reprend l'exemple du verbe *produire* (*to produce*). Ce verbe est du type *causatif physique* et appartient à la catégorie de verbes de *service* qui affecte l'objet, les buts exprimés par des verbes de *service*

sont nommés *convolutifs* pour indiquer que leur accomplissement est une convolution de processus, c'est-à-dire que les buts convolutifs représentent un *processus* de *processus*. Par exemple :

(*Produire*)_{verbe} (*une voiture*)_{objet}

La production de la voiture passe de la situation initiale « rien » jusqu'à la situation finale « construite ». Les états de la *voiture* changent :

voiture.situation_initial = 'rien' OU *voiture.situation_final* = 'construite' OU
voiture.situation_final = 'non-complète'.

3.2.2.5.1.1.2. Réaliser « raisonner sur les verbes causatifs physiques de processus » (DRI_{6.1.1.2})

La DRI_{6.1.1.2} permet de raisonner sur les états des verbes de type causatif physique de processus. Les verbes de *processus* expriment un développement par étapes au travers desquelles le paramètre *objet* change de *situation*. De ce fait, les verbes de *processus* sont caractérisés par l'acceptation des paramètres *situation* (*situation initiale* ou *situation finale* ou les deux). Les *situations* définissent des états et des circonstances du paramètre *objet* et ne sont pas des processus. Mais, les fonctions énoncées par le verbe de *processus* sont une simplification des fonctions énoncées par des verbes de *service*. Cela indique un affinement des fonctions des verbes de *service* en verbes de *processus*.

Les verbes de *service*, comme les verbes de *processus*, indiquent un développement par étapes impliquant le passage de l'objet d'une *situation initiale* à une *situation finale*. Le verbe de processus *traiter* (*to process*), par exemple, est un affinement du verbe de service *fournir* (*to supply*) du type *causatif physique*. Les buts énoncés par des verbes de *processus* sont appelés *progressifs*, afin d'indiquer que des manifestations des effets qu'ils causent se réalisent par étapes. Les buts *progressifs* affinent aussi les buts *convolutifs*. Par exemple :

(*Traiter*)_{verbe} (*la réservation*)_{objet}

La réservation passe de la situation initiale « début » jusqu'à la situation finale « accomplissement ». Les états de la *réservation* changent :

réservation.état_situation = 'rien' OU *réservation.situation_final* = 'terminée' OU
réservation.situation_final = 'reportée'.

3.2.2.5.1.1.3. Réaliser « raisonner sur les verbes causatifs physiques d'activité » (DRI_{6.1.1.3})

La DRI_{6.1.1.3} permet de raisonner sur les états des verbes de type causatif physique d'activité. Les verbes d'*activité* expriment un ensemble d'*actions* se réalisant en une seule étape. Cela signifie que les verbes de cette catégorie ne considèrent pas le changement de *situation* de leurs objets. Autrement dit, les verbes d'activité n'acceptent pas les paramètres *situation* (*situation initiale* ou *situation finale*). Tous les autres paramètres sont acceptés par les verbes d'*activité*.

Le verbe d'activité *installer* (*to install*), par exemple, est un *affinement* du verbe de processus *traiter* (*to process*) du type *causatif physique*. Les buts énoncés par des verbes d'*activité* sont appelés *stationnaires*. Ces buts indiquent que des *effets* qu'ils causent, se réalisent en une seule étape. Les buts *stationnaires* affinent les buts *progressifs*. A titre d'exemple on présente le but *stationnaire* suivant :

(*Installer*)_{verbe} (*la boîte de vitesse*)_{objet} *par* (*assemblage*)_{manière}

3.2.2.5.1.1.4. Réaliser « raisonner sur les verbes causatifs physiques d'action » ($DRI_{6.1.1.4}$)

La $DRI_{6.1.1.4}$ permet de raisonner sur les états des verbes de type causatif physique d'action. Les verbes d'action expriment un fait visant à produire un effet donné. En conséquence les verbes d'action ne s'occupent pas du changement de *situation* de leurs objets. Cela signifie que les verbes d'action n'acceptent pas les paramètres *situation* (*situation initiale* ou *situation finale*). Tous les autres paramètres sont acceptables.

L'exclusion des paramètres *situation* indique que les verbes d'action, ainsi que ceux d'activité, n'énoncent pas une réalisation progressive de leurs actions. Cette absence du paramètre *situation* donne le caractère *instantané* aux actions et aux états formulés par les verbes.

Par exemple, le verbe d'action *fixer* (*to set*) est un *affinement* du verbe d'activité *installer* (*to install*), du type *causatif physique*. Les buts énoncés par des verbes d'action sont appelés *buts instantanés*. Ceux-ci n'expriment pas des changements de *situations*. Ces buts constituent une expression instantanée des *effets* qu'ils causent. Par exemple :

(*Fixer*)_{verbe} (*la boîte de vitesse*)_{objet}

3.2.2.5.1.2. Réaliser « raisonner sur les verbes causatifs intellectuels » ($DRI_{6.1.1}$)

La $DRI_{6.1.2}$ permet de raisonner sur les états des verbes de type causatif intellectuel. Cette directive est de type choix. Elle offre quatre possibilités comme le montre la Figure 4.9 :

- $DRI_{6.1.2.1}$ pour raisonner sur les verbes de type causatif intellectuel de service
- $DRI_{6.1.2.2}$ pour raisonner sur les verbes de type causatif intellectuel de processus
- $DRI_{6.1.2.3}$ pour raisonner sur les verbes de type causatif intellectuel d'activité
- $DRI_{6.1.2.4}$ pour raisonner sur les verbes de type causatif intellectuel d'action

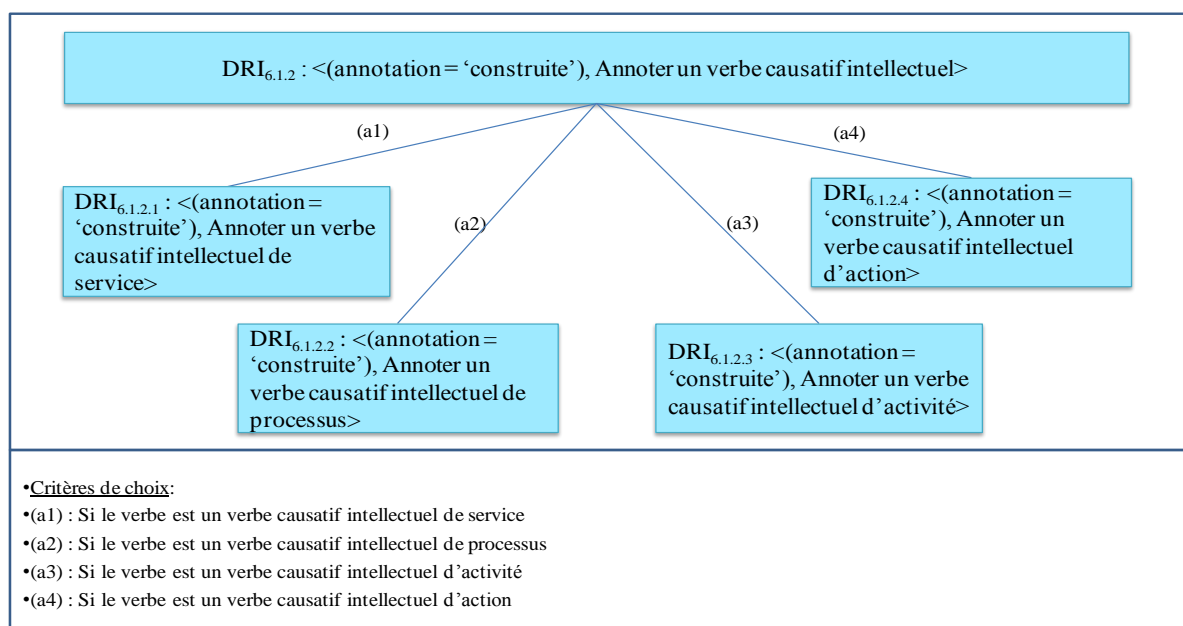


Figure 4.9. Structure de $DRI_{6.1.2}$

3.2.2.5.2. Réaliser « raisonner sur les verbes descriptifs » (DRI_{6.2})

Une fois que le type de verbe est identifié comme descriptif, la DRI_{6.2} permet de raisonner sur les états. Cette directive est de type choix. Elle offre deux possibilités comme le montre la Figure 4.10 :

- DRI_{6.2.1} pour raisonner sur les verbes de type descriptif physique
- DRI_{6.2.2} pour raisonner sur les verbes de type descriptif intellectuel

La DRI_{6.2.1} permet de raisonner sur les états des verbes de type descriptif physique. Cette directive est de type choix. Elle offre quatre possibilités :

- DRI_{6.2.1.1} pour raisonner sur les verbes de type descriptif physique de service
- DRI_{6.2.1.2} pour raisonner sur les verbes de type descriptif physique de processus
- DRI_{6.2.1.3} pour raisonner sur les verbes de type descriptif physique d'activité
- DRI_{6.2.1.4} pour raisonner sur les verbes de type descriptif physique d'action

La DRI_{6.2.2} permet de raisonner sur les états des verbes de type descriptif intellectuel. Cette directive est de type choix. Elle offre quatre possibilités :

- DRI_{6.2.2.1} pour raisonner sur les verbes de type descriptif intellectuel de service
- DRI_{6.2.2.2} pour raisonner sur les verbes de type descriptif intellectuel de processus
- DRI_{6.2.2.3} pour raisonner sur les verbes de type descriptif intellectuel d'activité
- DRI_{6.2.2.4} pour raisonner sur les verbes de type descriptif intellectuel d'action

Les détails de ces dix directives ressemblent à ceux des verbes causatifs.

3.2.2.6. Réaliser « Annoter le service par résolution des conflits » (DRI₇)

La DRI₇ est associée à la section < *Annoter le service, Annoter le service, par résolution des conflits* > de la carte Map-Publication (Figure 4.4). Elle vise à réaliser l'intention *Annoter le service* en utilisant l'une des stratégies *par résolution des conflits de verbe, par résolution des conflits d'objet* ou bien *par résolution des conflits de relation verbe-objet*.

Afin d'assurer que le verbe et le produit du but du service sont compatibles, cette directive est mise en place avec l'idée d'utiliser les liens verbe-produit dans l'ontologie des produits. Elle prend à l'entrée un verbe et un produit, elle cherche s'il y a une relation définie dans l'ontologie des produits entre la classe de sens du verbe et l'objet ; si une relation existe, l'objet est relié au verbe avec le lien « *produitHasVerb* » et ils sont compatibles ; sinon (il n'y a pas de relation qui relie l'objet avec la classe de sens du verbe), on propose à l'utilisateur de modifier :

- le verbe en introduisant les classes de sens de verbes qui sont reliées à ce produit
- le produit en introduisant les produits qui acceptent la classe de sens de verbe.

- de créer une relation entre la classe de sens de verbe et le produit (Figure 4.10).

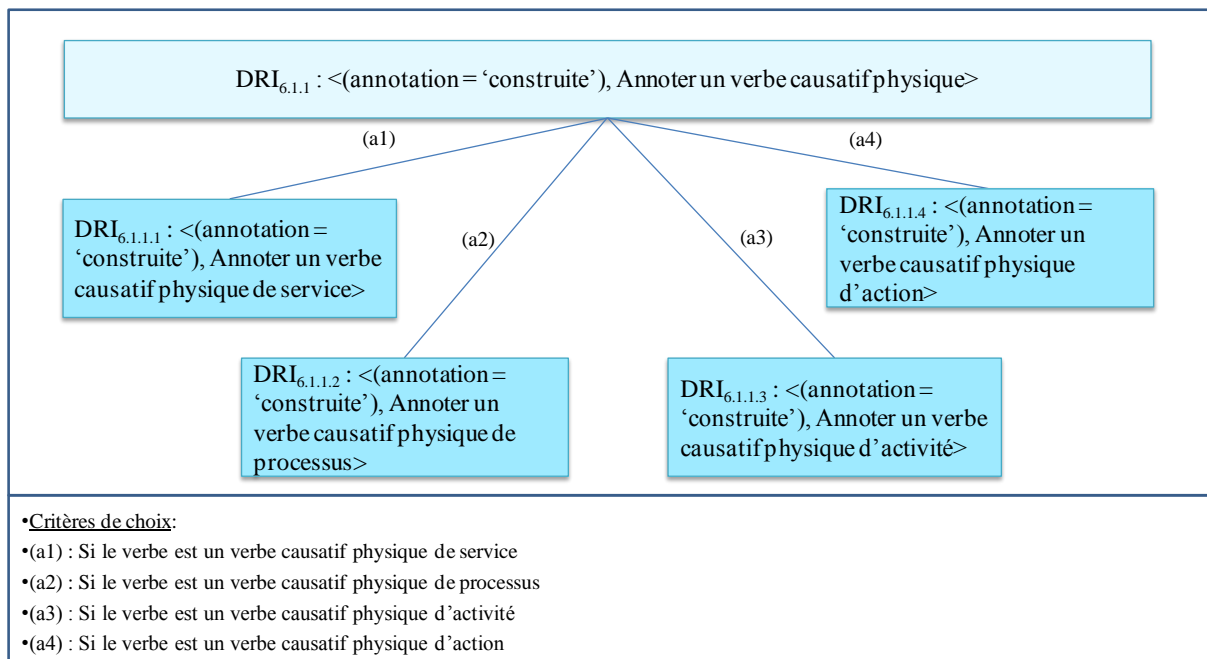


Figure 4.10. Structure de DRI_7

A la Figure 4.11, nous présentons un exemple qui montre la relation entre le produit et le verbe en utilisant l'ontologie des produits *pOnto*. Dans cette ontologie, nous avons ajouté un lien « *ProductHasverb* » à chaque objet concernant les verbes que ce produit accepte. Par exemple, le produit *patient* accepte les verbes appartenant à la classe de sens de verbe d'accueil « *recevoir, accueillir...* ».

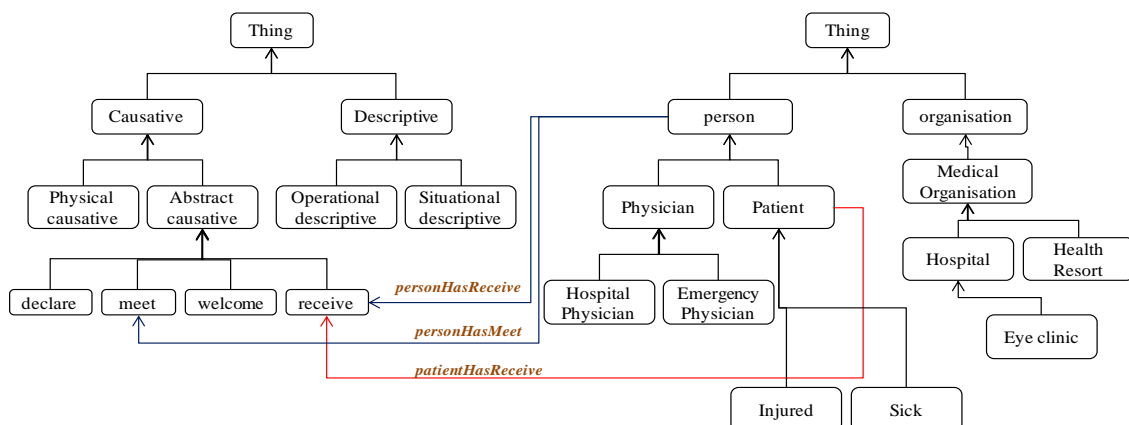


Figure 4.11. La résolution des conflits entre verbe et produit

Nous pouvons utiliser l'ontologie *pOnto* pour vérifier la compatibilité entre le verbe et le produit, on voit que le produit « *patient* » accepte le verbe « *recevoir* » qui appartient à la classe de sens de verbe « *accueil* ».

Au cas où le verbe n'appartient pas aux verbes que le produit accepte, nous pouvons résoudre ce conflit en suggérant au fournisseur les verbes qu'il peut utiliser avec ce produit.

3.2.3. Réaliser « Annoter le service Par vérification » (DRI_8)

La DRI_8 est associée à la section bc_1 de la carte Map-Publication (Figure 4.4). Elle vise à réaliser l'intention *Arrêter* en utilisant la stratégie *Par vérification*.

La directive DRI_8 vise à arrêter le processus de publication, c'est-à-dire l'exécution de la carte MAP-Publication, en vérifiant les annotations. Comme le montre la Figure 4.12, cette directive est de type plan et composée de deux sous-directives :

- $DRI_{8,1}$ pour vérifier les annotations.
- $DRI_{8,2}$ pour valider les annotations.

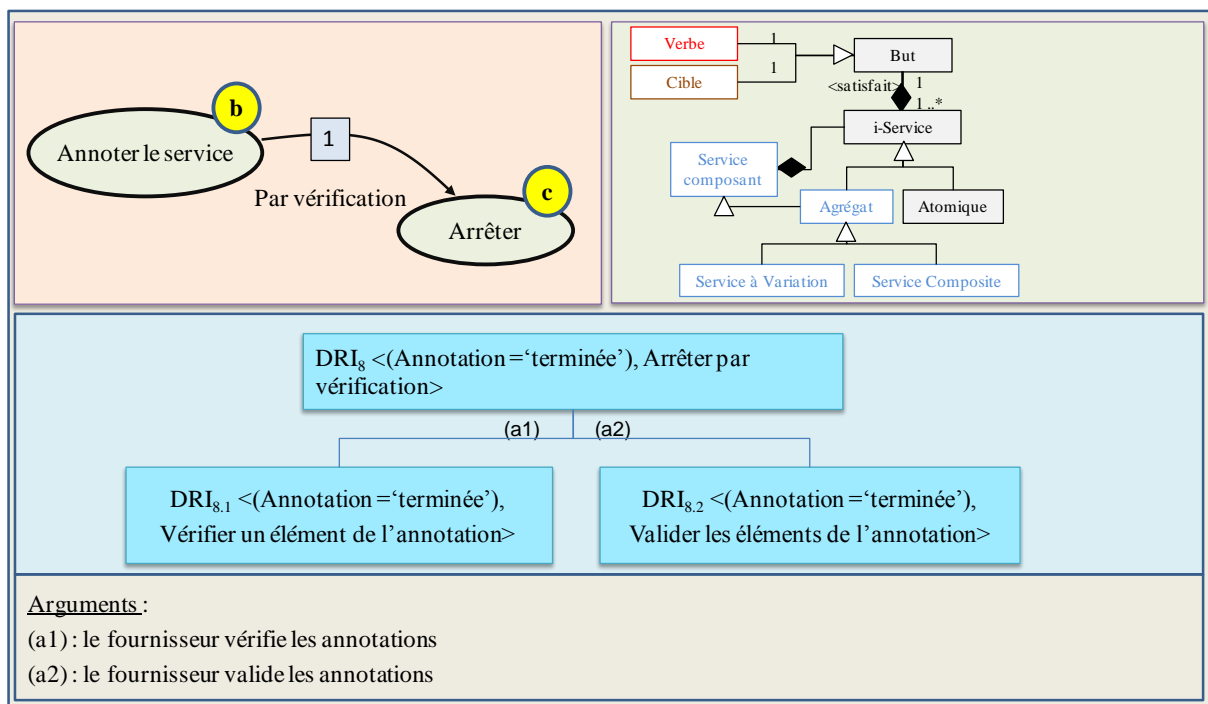


Figure 4.12. Structure de DRI_8

3.2.3.1. Réaliser « vérifier un élément de l'annotation » ($DRI_{8,1}$)

Cette directive vise à définir un ensemble de critères pour la vérification des annotations de services. La vérification des annotations consiste à s'assurer de la structure des annotations qu'on a utilisées en particulier, le verbe, l'objet, le type de service (atomique ou agrégat) qui sont les éléments essentiels.

Comme le montre la Figure 4.13, cette directive est de type plan et composée de trois sous-directives :

- $DRI_{8.1.1}$ pour vérifier le verbe.
- $DRI_{8.1.2}$ pour valider l'objet.
- $DRI_{8.1.3}$ pour valider le type de service.

Exécuter un plan revient à parcourir son graphe de précédence. Lors de l'exécution d'un contexte plan, les critères de choix des liens de précédence sont calculés grâce à l'évaluation de leurs arguments.

Dans la directive $DRI_{8.1}$, trois directives sont à exécuter. La directive $DRI_{8.1.1}$ vise à vérifier si le verbe existe et il est bien annoté. Ensuite la directive $DRI_{8.1.2}$ vise à vérifier si l'objet existe et il est bien annoté. Finalement, la directive $DRI_{8.1.3}$ vise à vérifier si le type de service (atomique ou agrégat) existe et il est bien annoté.

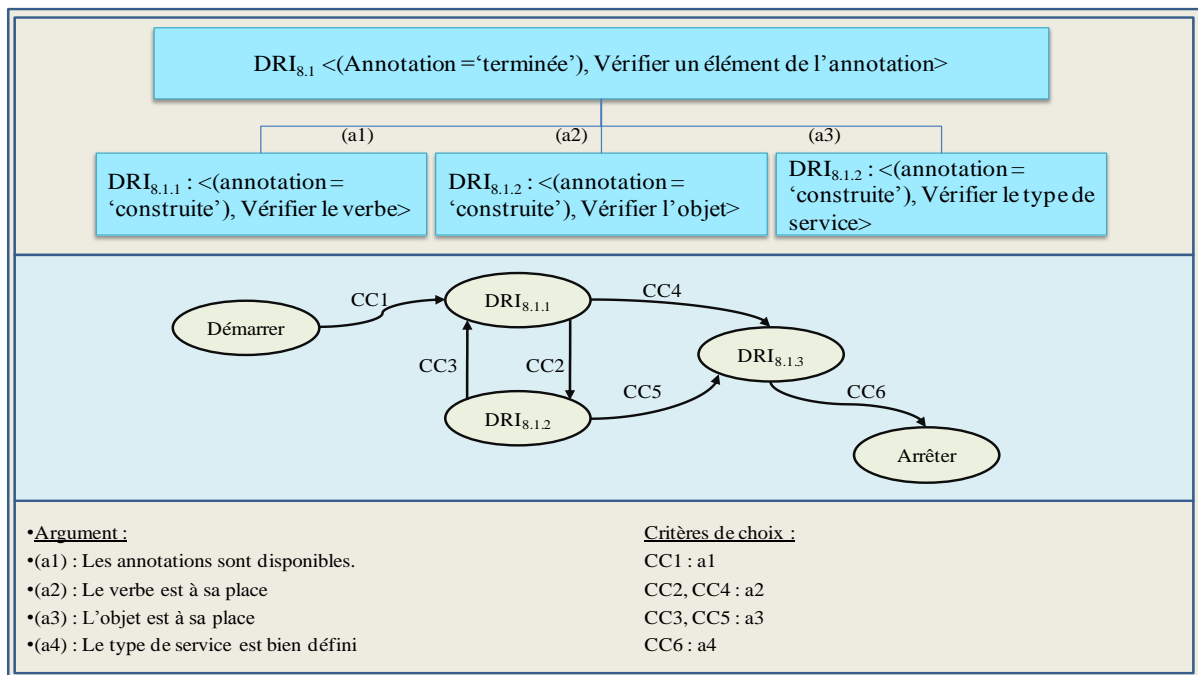


Figure 4.13. Structure de $DRI_{8.1}$

3.2.3.2. Réaliser « valider les annotations » ($DRI_{8.2}$)

Cette directive vise à valider les annotations de services. La validation des annotations consiste à accepter les valeurs des annotations après avoir terminé la structuration et l'annotation.

4. CONCLUSION

Le processus de guidage d'annotation introduit dans ce chapitre aide le fournisseur dans sa démarche de publication des services intentionnels. Ce processus enrichit les annotations des services pour les rendre faciles à retrouver.

Le processus de publication est un élément fondamental dans notre travail car la fonction de recherche et le moteur de recherche comptent sur l'annotation enrichie pour faire la correspondance avec les demandes de l'utilisateur et retrouver les services intentionnels pertinents.

Dans ce chapitre, nous avons utilisé l'ontologie de services intentionnels en tant que base pour ajouter des annotations sémantiques dirigées par le but à des services web. Ces annotations sémantiques sont utilisées pour définir les descripteurs des services intentionnels atomiques et agrégats. Le descripteur obtenu est nettement plus expressif que la recommandation du W3C et les présentations au W3C sur la sémantique des services Web.

CHAPITRE 5

RECHERCHE DE SERVICES ET FORMULATION DE REQUÊTES

1. INTRODUCTION

La recherche de services intentionnels est basée sur la correspondance entre la requête de l'utilisateur (exprimée sous forme de but) et les buts des services intentionnels disponibles dans l'annuaire. Ce chapitre est centré sur le guidage de l'utilisateur dans la formulation de sa requête et dans la recherche et l'exploration des services proposés.

Dans ce chapitre, nous utilisons le modèle de but avec une analyse linguistique. Toutefois, pour enrichir le but sémantiquement, nous proposons d'aider l'utilisateur à reformuler sa requête afin d'augmenter la possibilité de trouver des services intentionnels. Pour cette raison, nous proposons un processus de reformulation de requêtes qui aide l'utilisateur à enrichir sa requête afin de sélectionner les services les plus proches de ses besoins.

Ce chapitre est composé de cinq sections. Nous discutons le langage des requêtes à la section 2. Dans la section 3, nous présentons le processus de reformulation. Dans la section 4, nous présentons un scénario d'utilisation. Finalement, nous concluons le chapitre par la section 5.

2. LE LANGAGE DE REQUETES

Le langage de requêtes des utilisateurs désigne la façon par laquelle les utilisateurs d'un service formulent leurs besoins d'une manière compréhensible de la part de l'annuaire des services. Dans des travaux connexes, la requête d'un utilisateur pour la satisfaction de ses besoins peut prendre plusieurs formes, selon le choix du fournisseur : (i) le fournisseur met à la disposition de l'utilisateur un mécanisme d'interrogation, (ii) l'utilisateur exprime sa requête dans le langage de description de service et l'envoie directement à l'annuaire, qui fait correspondre la demande à l'offre via un algorithme d'appariement ou bien (iii) l'utilisateur

exprime sa requête dans le langage naturel. Dans le troisième cas, la requête de l'utilisateur est exprimée par mots-clés ou en langage naturel libre.

Sur le Web, la recherche par mot-clé a été popularisée par les moteurs de recherche comme Google. Lorsqu'on recherche un document ou une page web contenant une partie spécifique du contenu exprimé par les mots-clés de recherche, ces moteurs sont généralement satisfaisants dans la plupart des cas. Cependant, quand il s'agit de la recherche d'un service web (ou un composant logiciel), la recherche par mots-clés ne peut pas donner des résultats satisfaisants car la sémantique d'un service web est plus complexe et difficile à capturer par un seul ou une combinaison de mots [Papazoglou et *al.*, 2007] [Al-Masri et *al.*, 2008]. Nous soutenons que cette combinaison de mots doit être analysée sémantiquement de sorte que la distinction entre le verbe, la cible et les différents types de paramètres soit faite. Par exemple, la recherche d'un programme FTP, la requête doit inclure le verbe « transférer » qui exprime une action, et l'objet « fichier » qui exprime l'objet de l'action.

Dans cette thèse, nous nous intéressons aux requêtes des utilisateurs exprimées à l'aide du formalisme de buts. Les modèles des besoins et en particulier ceux dirigés par les buts sont plus facilement compréhensibles par des utilisateurs d'un domaine métier que les modèles de services qui exigent des connaissances techniques.

Pour interpréter la requête d'un utilisateur exprimée sous forme de but, nous faisons appel à une solution à base d'ontologie pour l'enrichissement, la reformulation et l'interprétation de cette requête. Les ontologies fournissent un vocabulaire commun pour spécifier les services. Elles contiennent un ensemble de termes relatifs à un domaine, pouvant être exploitées pour exprimer (sous forme de requêtes) les besoins des utilisateurs.

Le requêtes, dans notre travail, sont les buts de l'utilisateur. Dans la suite, nous présentons quelques exemples des buts qui représentent les requêtes d'un utilisateur :

- (Examiner)_{Verbe} (un patient) _{Objet}
- (Générer)_{Verbe} (un rapport médical) _{Résultat} pour (le patient) _{bénéficiaire}
- (Livrer)_{Verbe} des (médicaments) _{Objet} de la (pharmacie) _{Source} (automatiquement) _{Manière}
- (Payer)_{Verbe} la (réservation) _{Objet} par (carte bancaire) _{Manière}
- (Fournir)_{Verbe} (des résultats de radiologie) _{Résultat} à nos (patients) _{Référent} par (email) _{Moyen}

3. LA REFORMULATION ET LE DEGRE DE SIMILARITE

La mise en correspondance entre les éléments du but de l'utilisateur et les éléments de l'intention de service est réalisée en comparant sémantiquement ces éléments un par un. Cette comparaison est réalisée à l'aide d'ontologies, ce qui nous permet de faire cette comparaison au moment de la recherche. Dans ce cas, l'implémentation d'un algorithme de recherche nécessite la modification des algorithmes existants dans un annuaire de services. Ainsi, cette modification exige l'intervention au niveau physique de l'annuaire afin de réaliser un nouveau moteur de recherche.

Contrairement, la reformulation de la requête nous permet de réaliser cette comparaison au niveau conceptuel avant de lancer la recherche. D'autres avantages : la reformulation (i) donne à l'utilisateur la possibilité d'interagir à chaque moment qu'il trouve que la reformulation diverge de ce qu'il a voulu chercher, (ii) permet d'éviter la modification des

algorithmes de recherche existants, et (iii) permet l'enrichissement de la requête au moment de la formulation.

Cependant, le résultat de la reformulation d'une requête est une liste de nouvelles formulations, cette liste exige d'être triée afin de définir une priorité entre ces formulations. Pour cette raison, nous proposons d'associer à chaque formulation un degré de similarité. Ce degré de similarité mesure la distance sémantique entre les éléments dans la requête originale de l'utilisateur et les éléments de la nouvelle formulation. Nous utilisons les relations entre les concepts dans une ontologie pour définir ce degré de similarité, en particulier, l'ontologie des produits (qui est une ontologie de domaine) qui contient les relations de généralisation et de spécialisation.

Dans la suite, nous présentons un algorithme pour la mesure du degré de similarité et un exemple d'application de cet algorithme.

3.1. L'algorithme d'appariement

Dans cette section, nous détaillons l'algorithme d'appariement. Cet algorithme est utilisé pour mesurer le degré de similarité entre les concepts de l'ontologie des produits. Cette ontologie est une ontologie de domaine et normalement, elle contient plusieurs types de relations entre ses concepts. Mais, nous nous intéressons seulement aux liens de généralisation « *isA* » et de spécialisation « *has* », parce que ces liens sont équivalents à la hyperonymie/hyponymie dans le dictionnaire, et donnent des sens proches du celui du concept recherché. Tandis que, les autres liens (tels que « *works for* »...) donnent d'autres informations et sont sémantiquement moins importantes.

Pour cette raison, nous introduisons le degré de similarité comme une distance sémantique entre deux concepts. Nous définissons la distance sémantique une valeur entre 0 et 100 qui représente à quel point deux concepts sont proches sémantiquement.

Le principe est simple : il est d'utiliser deux paramètres α et β qui représentent la distance sémantique entre un concept et sa généralisation ou sa spécialisation respectivement, Figure 5.1.

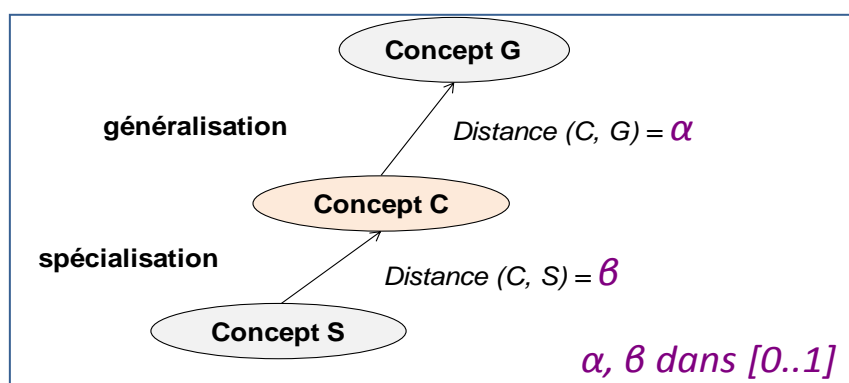


Figure 5.1. Le degré de similarité

Nous définissons le degré de similarité entre deux concepts comme une distance sémantique entre eux.

Pour chaque concept C : $Dist(C, C_g) = \alpha$ où C_g est la généralisation de C .

$Dist(C, C_s) = \beta$ où C_s est une spécialisation de C .

L'algorithme d'appariement prend en entrée :

- un *concept* qui représente l'objet du but formulé par la requête de l'utilisateur,
- une *ontologie*, c'est l'ontologie des produits (l'ontologie de domaine), et
- un *seuil* (*threshold*) qui prend sa valeur dans l'intervalle $[0,1]$: il définit le degré de similarité minimal accepté pour un concept pour qu'il apparaisse dans le résultat.

L'algorithme produit à la sortie, une liste de couples (concept, degré de similarité) :

- le *concept* représente le concept trouvé dans l'ontologie et qui est adjacent au concept d'entrée, et
- le *degré de similarité* calculé et associé à ce concept.

Dans cet algorithme, nous définissons deux paramètres α et β qui prennent leurs valeurs dans l'intervalle $[0,1]$:

- α représente la distance sémantique entre un concept et sa généralisation (la classe supérieure) et,
- β représente la distance sémantique entre un concept et sa spécialisation (la classe inférieure).

L'algorithme se déroule de la façon suivante :

- ✓ l'algorithme localise le concept d'entrée dans l'ontologie, et lui associe le degré de similarité égal à 100.
- ✓ l'algorithme cherche si ce concept a un parent (lien de généralisation) ; dans le cas où il trouve un parent, il lui associe le degré de similarité égal au degré de similarité du concept en cours * α . Si le nouveau degré est supérieur au seuil, on ajoute le concept à notre liste avec son degré de similarité associé.
- ✓ l'algorithme cherche si ce concept a un fils (lien de spécialisation) ; dans le cas où il trouve un fils, il lui associe le degré de similarité égal au degré de similarité du concept en cours * β . Si le nouveau degré est supérieur au seuil, on ajoute le concept à notre liste avec son degré de similarité associé.
- ✓ Si la liste contient des concepts non examinés, nous répétons les règles 2 et 3 pour chaque concept.
- ✓ L'algorithme s'arrête quand tous les concepts de la liste ont été examinés.

Dans la Figure 5.2, nous présentons l'algorithme de mesure du degré de similarité.

Entrée : concept, ontologie de domaine, seuil

Initialiser α /* degré de similarité avec la généralisation du concept en question

Initialiser β /* degré de similarité avec la spécialisation du concept en question

Degré de similarité = 1

Liste des concepts adjacents = \emptyset

Si *concept* appartient à l'ontologie alors

Ajouter *concept* avec son degré de similarité qui égal à 1 dans la *liste des concepts adjacents*

J = 1

Pour chaque *concept j* non examiné dans la *liste des concepts adjacents* faire

Si *concept j* a un parent alors

Degré de similarité du parent = α * Degré de similarité du concept

Si Degré de similarité du parent > seuil alors

Ajouter le parent avec son degré de similarité dans la liste

Si *concept j* a un fils alors

Degré de similarité du fils = β * Degré de similarité du concept

Si Degré de similarité du fils > seuil alors

Pour chaque fils du concept en cours (non existant dans la liste)

Ajouter le fils avec son degré de similarité dans la liste

Marquer *concept j* comme examiné

J++

Afficher liste des concepts adjacents

Figure 5.2. *Algorithme d'appariement*

3.2. Exemple d'utilisation de l'algorithme d'appariement

Dans l'exemple de la Figure 5.3, on applique l'algorithme à cette ontologie, en prenant en entrée le concept « *Médecin d'hôpital* » et un seuil de 0.5. Nous fixons les paramètres $\alpha = 0.85$ et $\beta = 0.75$.

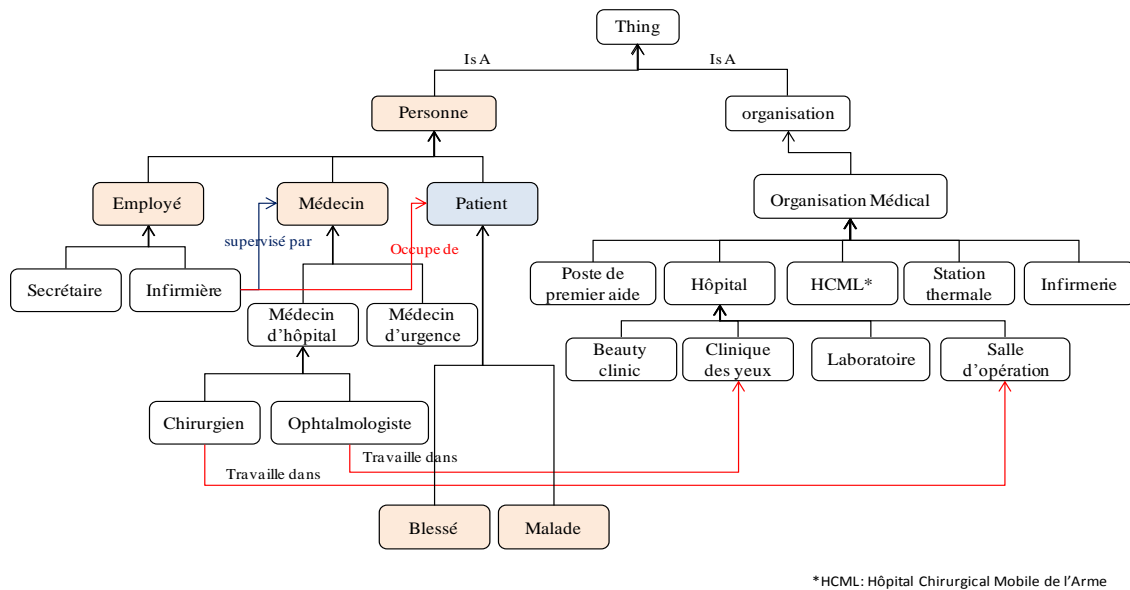


Figure 5.3. Un aperçu de l'ontologie des produits dans le domaine de la santé [Klisch et al., 2006]

Nous aurons comme résultat les couples présentés dans la Figure 5.4. Les concepts acceptés dans la liste sont marqués en vert avec les degrés d'appariement en gris, les chiffres en orange montrent la séquence des concepts examinés. Les croix rouges indiquent les concepts non acceptés dans la liste à cause du dépassement de seuil.

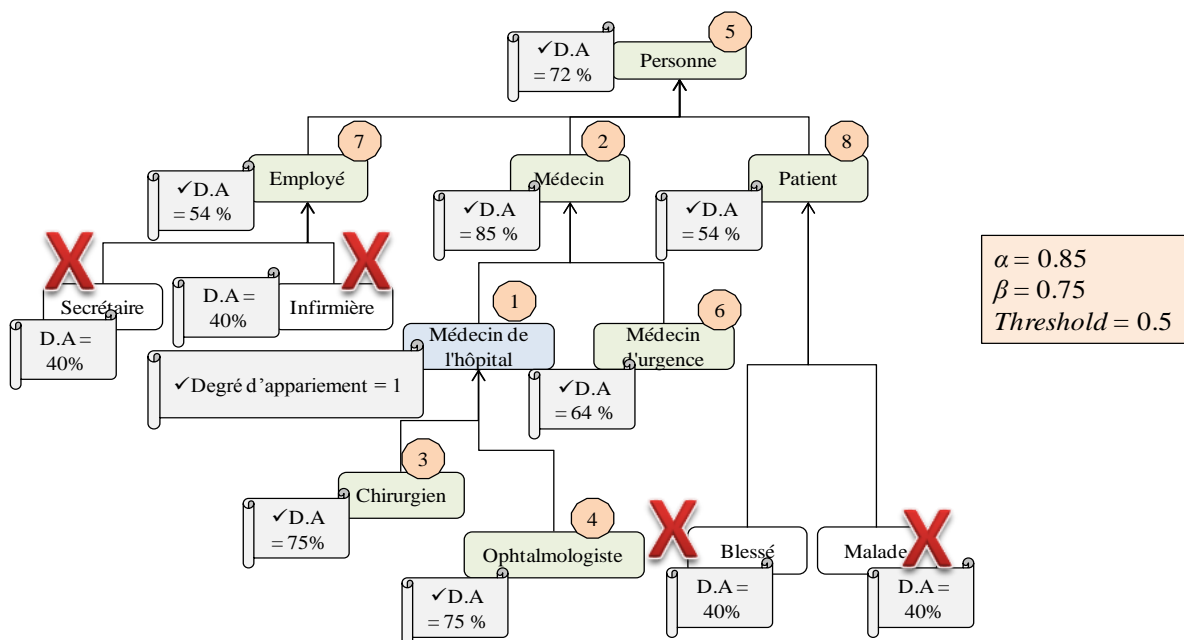


Figure 5.4. Le résultat de l'algorithme pour (« Médecin d'hôpital », « Ontologie de Santé », « Threshold = 0.5 »)

4. PROCESSUS DE REFORMULATION DE REQUETES

Dans ce paragraphe, nous proposons un processus de reformulation de requêtes. Ainsi que les directives de l'utilisation de ce processus. Ce processus va aider l'utilisateur à enrichir sa requête afin de sélectionner les services les plus proches de ses besoins.

La Map-Recherche présentée à la Figure 5.5 définit, outre les intentions **Démarrer** et **Arrêter** que l'on retrouve dans toute carte, les deux intentions clés à atteindre : **formuler une requête** qui correspond à l'enrichissement de la requête de l'utilisateur, et **proposer des services** qui correspond au résultat de la recherche. Ce processus est modalisé en utilisant le formalisme de la Carte.

L'intention **formuler une requête** recouvre les composantes du processus qui concernent la reformulation de la requête de l'utilisateur.

L'intention **proposer des services** recouvre les composantes du processus qui concernent la recherche, l'exploration et l'affichage des services résultats.

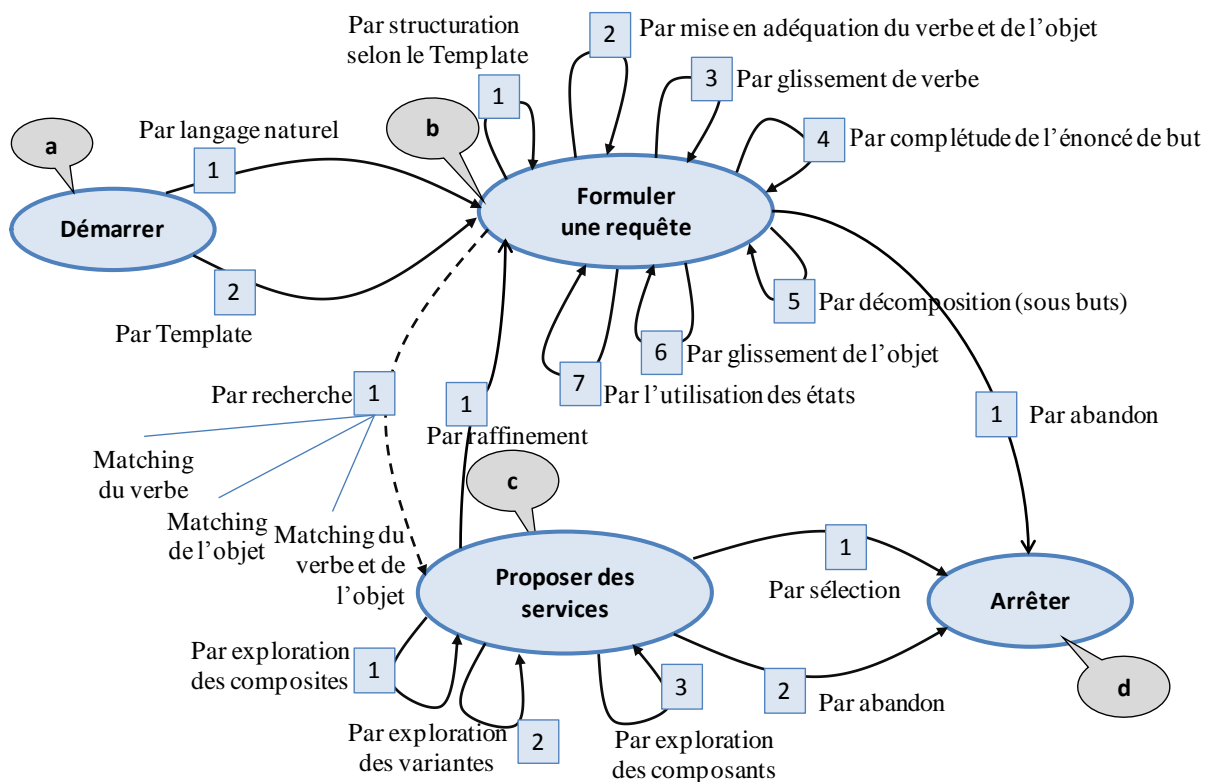


Figure 5.5. *Processus de recherche des services et de formulation de requêtes (Map-Recherche)*

Les 17 sections qui composent le Map-Recherche, proposent des stratégies pour la réalisation de ses quatre intentions.

- La stratégie **Par langage naturel** : cette stratégie a été introduite dans la perspective de capturer les éléments constituant les buts des utilisateurs formulés en langage naturel. Les utilisateurs qui préfèrent cette stratégie ne connaissent ni le formalisme de but, ni comment ils peuvent s'en servir. Ils préfèrent chercher les services

intentionnels de la même façon qu'ils utilisent les moteurs de recherche sur internet (Google...), où ils entrent leurs textes libres ou une combinaison de mots clés. Cependant, cette stratégie offre à ces utilisateurs de reformuler leurs requêtes en supposant que l'utilisateur dans son texte, saisit le verbe d'abord puis l'objet. La sortie de cette stratégie est une reformulation basée sur l'hypothèse précédente.

- La stratégie ***Par Template*** : cette stratégie, alternative à la première, a été introduite pour saisir les éléments constituant le but ou une partie de but exprimé dans la requête de l'utilisateur. Les utilisateurs de cette stratégie sont capables d'identifier les éléments de but (verbe, objet, ...), où ils peuvent saisir ces éléments dans un template de but prédéfini pour cet objectif. Ce template est basé initialement sur le modèle de but. La sortie de cette stratégie est la requête de l'utilisateur structurée selon le modèle de but. Dans cette structuration, les éléments constituant le but sont identifiés.
- La stratégie ***Par structuration selon le Template*** : cette stratégie a été introduite pour aider l'utilisateur à structurer sa requête selon le template de but. Dans le cas où l'utilisateur saisit sa requête en langage naturel et sans prendre compte de la structuration, cette stratégie peut-être utilisée pour structurer les éléments de la requête selon le template de but. Donc, cette stratégie prend à l'entrée une requête non structurée, écrite en langage naturel, et rend à la sortie une requête où les composants sont structurés selon le modèle but.
- La stratégie ***Par mise en adéquation du verbe et de l'Objet*** : cette stratégie a été introduite pour faire en sorte que le verbe et l'objet de la requête de l'utilisateur sont compatibles. Cette stratégie est mise en œuvre avec l'idée d'utiliser les liens verbe-produit dans l'ontologie des produits. Elle prend en entrée un verbe et un produit ; elle cherche s'il y a une relation définie dans l'ontologie des produits entre la classe de sens du verbe et le produit ; si une relation existe, le produit est relié au verbe avec le lien « *produitHasVerbe* » et ils sont compatibles. Dans le cas où il n'y a pas de relation qui relie le produit avec la classe de sens du verbe, cette stratégie propose à l'utilisateur de modifier le verbe et introduit les classes de sens de verbe qui sont reliées à ce produit.
- La stratégie ***Par glissement de verbe*** : cette stratégie permet de remplacer le verbe dans la requête de l'utilisateur. Le remplacement se fait par un autre verbe plus fréquent appartenant à la même classe de sens de verbe. Une classe de sens de verbe est une sous-classe des seize classes rencontrées dans l'ontologie des verbes. On utilise cette stratégie d'abord, pour s'assurer du sens de verbe, parce qu'un verbe normalement appartient à une ou plusieurs classes de sens de verbes. Par conséquent, cette stratégie aide l'utilisateur aussi, dans la précision de la classe de sens de verbe, en donnant plus d'information et des exemples d'utilisation.
- La stratégie ***Par complétude de l'énoncé de but*** : cette stratégie nous permet de compléter la déclaration du but par l'exécution des règles de complétude associées à chaque classe de verbes. Tout dépend de la classe de sens de verbe ; chaque classe de sens de verbe est une sous-classe de l'une des seize classes rencontrées dans l'ontologie des verbes [Urrego, 2005]. Dans l'ontologie des verbes, un verbe appartient à une classe, et à partir de cette classe, nous aurons la signature de la classe. La signature d'une classe de verbes donne plusieurs informations. C'est la catégorie de verbe qui implique la caractérisation des paramètres en termes de paramètres *facultatifs* ou *interdits*.
- La stratégie ***Par décomposition (sous buts)*** : cette stratégie concerne la découverte d'un sous-but dans la manière du but initial. Ceci implique la décomposition de ce but en deux sous-buts, le premier est le but initial et le deuxième et le but trouvé dans la manière.

- La stratégie **Par glissement de l'objet** : cette stratégie implique la recherche de concepts voisins dans l'ontologie des produits, et de remplacer le produit avec d'autres concepts similaires. Dans cette stratégie, on prend le concept d'objet, on le localise dans l'ontologie des produits qui rend tous les concepts adjacents à cet objet. Pour cette raison, nous avons défini un degré de similarité au paragraphe 3.2. Ce degré de similarité sert à définir la distance sémantique entre le concept d'origine et les adjacents dans l'ontologie des produits. Ainsi, nous pouvons proposer à l'utilisateur d'autres concepts, chacun portant un degré de similarité avec l'original.
- La stratégie **Par l'utilisation des états** : cette stratégie permet de spécifier le changement des états de buts. Ainsi, l'exécution d'une requête peut être considérée comme le passage d'un *état initial* vers un *état final* résultant de la réalisation de but. Cette stratégie dépend du type de verbe qui indique le type d'intervention *directe* ou *indirecte* sur le paramètre *objet*. Il indique si l'intervention est *physique* ou *intellectuelle* sur le paramètre *objet*.
- La stratégie **Par recherche** : cette stratégie est utilisée pour lancer la recherche de services candidats qui correspondent à la requête initiale reformulée par l'utilisateur. Pour une requête donnée, nous transformons les éléments de la requête au format d'un descripteur de service intentionnel, puis nous lançons la recherche. Nous pouvons utiliser les algorithmes standards d'un annuaire de services sémantiques. La sortie de cette stratégie est un ensemble d'ensembles de services, dont la réponse à chaque reformulation est un sous-ensemble. Le résultat total est normalement trié, parce que nous avons exécuté les reformulations en ordre.
- La stratégie **Par affinement** : cette stratégie est utilisée pour affiner la requête originale de l'utilisateur. C'est le cas où la reformulation n'aboutit à rien ou les services trouvés sont loin des besoins de l'utilisateur. L'entrée de cette stratégie est une requête modifiée. La modification concerne le verbe ou le produit ou les paramètres ou tous ces aspects en même temps. La sortie de cette stratégie est une requête qui respecte le modèle de but.
- La stratégie **Par affichage des résultats** : cette stratégie assure l'affichage du résultat de la recherche de services. L'affichage est trié selon le degré de similarité de chaque service trouvé, évidemment, le degré affiché reflète le degré de similarité de la reformulation utilisée avec la requête initiale.
- La stratégie **Par exploration des composites** : cette stratégie est utilisée pour explorer le service composé. C'est dans le cas où le service est un composant d'un ou plusieurs services de type composite. Dans ce cas, l'utilisateur a le choix de sélectionner le service qu'il trouve approprié à ces besoins.
- La stratégie **Par exploration des composants** : cette stratégie est utilisée pour explorer les composants d'un service agrégat de type composite. Dans cette stratégie, nous pouvons explorer les composants d'un service pour proposer à l'utilisateur le choix de sélectionner le service qu'il trouve approprié à ses besoins.
- La stratégie **Par exploration des variantes** : cette stratégie est utilisée pour explorer les composants d'un service agrégat de type à variation. Dans cette stratégie, nous pouvons proposer à l'utilisateur le choix de sélectionner le service qu'il trouve approprié à ses besoins.
- La stratégie **Par sélection** arrête le processus dans le cas où l'utilisateur sélectionne un ou les services qui répondent à ses besoins.
- La stratégie **Par abandon** arrête le processus dans le cas où une décision serait prise par l'utilisateur et dans ce cas le processus est terminé.

4.1. Sommaire des directives du Map-Recherche

Conformément au modèle du MAP, chaque directive de type DSI a une signature composée d'une situation et d'une intention. De plus, un identifiant unique est systématiquement employé pour rapidement désigner les DSI. L'identifiant est de la forme DSI_n , où n est un numéro d'intention. La directive DSI_n est définie pour guider la sélection de la prochaine intention à réaliser sachant que l'intention I_n a déjà été elle-même réalisée.

Map-Recherche a quatre intentions : « Démarrer », « formuler une requête », « proposer des services » et « Arrêter ». L'intention « Arrêter » mise à part, les intentions du Map-Recherche appartiennent au moins à deux sections ayant deux intentions cibles différentes. Cependant, Map-Recherche (Figure 5.5) propose deux stratégies pour progresser de l'intention source « Démarrer » vers l'intention cible « formuler une requête ». Mais, pour progresser de « formuler une requête » vers « formuler une requête », il propose sept stratégies, une stratégie de « formuler une requête » vers « proposer des services », trois stratégies de « proposer des services » vers « proposer des services », une stratégie de « proposer des services » vers « formuler une requête », deux stratégies de « proposer des services » vers « Arrête » et une stratégie de « formuler une requête » vers « Arrête ». Par conséquent, comme le montre le Tableau 5.1, Map-Recherche a trois DSS. Ce DSS est référencé par un identifiant unique de type DSS_n où n est un numéro séquentiel unique pour les DSS du MAP. Une notation pointée est employée pour garantir l'unicité globale des numéros de DSS.

Le Map-Recherche contient dix-sept sections. A chacune de ces sections une DRI est associée. Les dix-sept DRI correspondant aux sections du Map-Recherche sont identifiées au Tableau 6.1. Tout comme les DSS, les DRI sont référencées au moyen d'un identifiant de type DRI_n où n est unique relativement au MAP concerné.

Le Tableau 6.1 comprend trois types de lignes faisant correspondre, la DSI, la DSS et les DRI référencées par cette DSS. Ainsi, trois colonnes faisant correspondre, pour chaque directive son type et la sous-section qui la détaille dans ce chapitre.

Tableau 6.1. Les directives du Map-Recherche

Les sections de la carte Map-Recherche	Type	Section
Progresser vers Formuler une requête	DSS_1	4.2.1
• ab1 : < Démarrer, Formuler une requête, Par langage naturel>	DRI_1	4.2.2.1
• ab2 : < Démarrer, Formuler une requête, Par Template>	DRI_2	4.2.2.2
Progresser depuis Formuler une requête	DSI_1	4.2.2
• bc1 : < Formuler une requête, Proposer des services, Par recherche>	DRI_3	4.2.2.1
• bd1 : < Formuler une requête, Arrêter, Par abandon>	DRI_4	4.2.2.2
Progresser vers Formuler une requête	DSS_2	4.2.2.3
• bb1 : < Formuler une requête, Formuler une requête, Par structuration selon le Template>	DRI_5	4.2.2.3.1
• bb2 : < Formuler une requête, Formuler une requête, Par mise en adéquation du verbe et de l'objet>	DRI_6	4.2.2.3.2
• bb3 : < Formuler une requête, Formuler une requête, Par glissement de verbe>	DRI_7	4.2.2.3.3

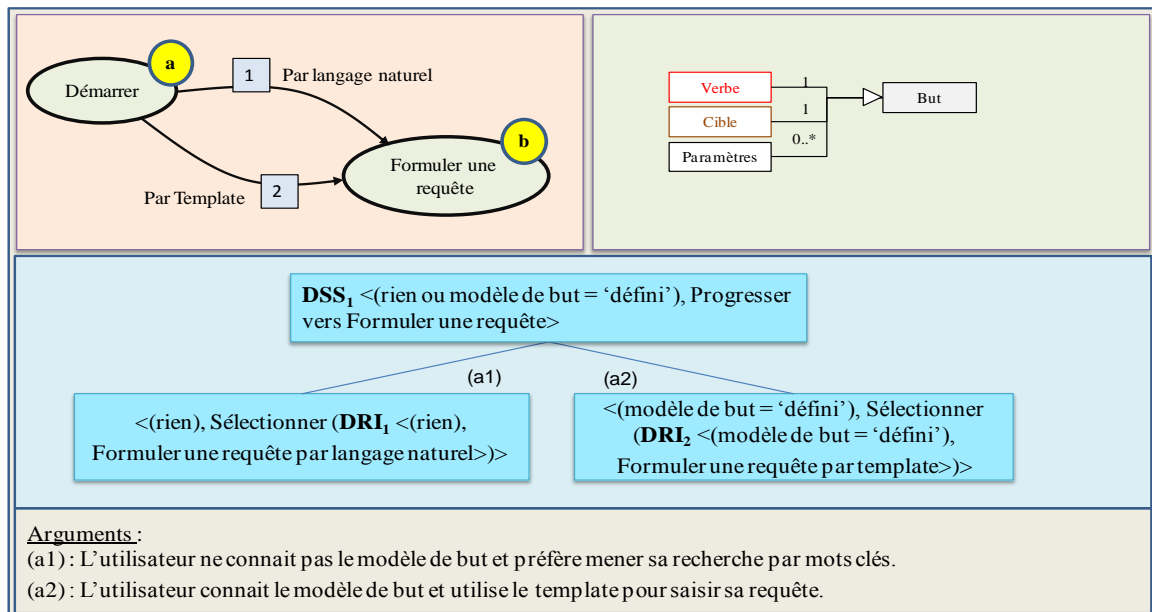
• bb4 : < Formuler une requête, Formuler une requête, Par complétude de l'énoncé de but>	DRI ₈	4.2.2.3.4
• bb5 : < Formuler une requête, Formuler une requête, Par décomposition (sous buts)>	DRI ₉	4.2.2.3.5
• Bb6 : < Formuler une requête, Formuler une requête, Par glissement de l'objet>	DRI ₁₀	4.2.2.3.6
• Bb7 : < Formuler une requête, Formuler une requête, Par l'utilisation des états>	DRI ₁₁	4.2.2.3.7
Progresser depuis Proposer des services	DSI ₂	4.2.3
• cb1 : < Proposer des services, Formuler une requête, Par affinement>	DRI ₁₂	4.2.3.1
Progresser vers Proposer des services	DSS ₃	4.2.3.2
• cc1 : < Proposer des services, Proposer des services, Par exploration des composites >	DRI ₁₃	4.2.3.2.1
• cc2 : < Proposer des services, Proposer des services, Par exploration des variantes >	DRI ₁₄	4.2.3.2.2
• cc3 : < Proposer des services, Proposer des services, Par exploration des composants >	DRI ₁₅	4.2.3.2.3
Progresser vers Arrêter	DSS ₄	4.2.3.3
• cd1 : < Proposer des services, Arrêter, Par sélection>	DRI ₁₆	4.2.3.3.1
• cd2 : < Proposer des services, Arrêter, Par abandon>	DRI ₁₇	4.2.3.3.2

4.2. Exploration du Map-Recherche

Dans cette section, nous présentons les directives DSI, DSS et DRI identifiées à la section 4.1. La présentation des directives est regroupée suivant la progression dans le Map. Les principales sous sections sont décrites suivant la progression depuis « *Démarrer* », « *Formuler une requête* » et « *Proposer des services* ».

4.2.1. Progresser vers « *Formuler une requête* » (DSS₁)

Le Map-Recherche indique qu'à partir de la situation initiale, il est possible de progresser en sélectionnant la seule intention « *Formuler une requête* ». Par conséquent, la DSS₁ guide la sélection de l'une des deux stratégies permettant de progresser vers « *Formuler une requête* » à partir de « *Démarrer* » (Figure 5.6 – partie supérieure).

Figure 5.6. La structure de DSS₁

Il s'agit d'une directive choix proposant deux alternatives :

- Sélectionner DRI₁ <(rien), Formuler une requête Par langage naturel > ,
- Sélectionner DRI₂ <(rien), Formuler une requête Par template>.

Les deux critères de choix (a1) et (a2) associés à la DSS₁ montrent que le premier choix est judicieux dans le cas où l'utilisateur ne connaît pas le modèle de but. Dans ce cas, peu d'importance est donnée dans un premier temps à la structure de but. Le deuxième choix est considéré quand l'utilisateur connaît le modèle de but (Figure 5.6- partie inférieure). Les arguments sont donc :

- (1) L'utilisateur ne connaît pas le modèle de but et préfère mener sa recherche par mots clés.
- (2) L'utilisateur connaît le modèle de but et utilise le template pour saisir sa requête.

Il est à noter que l'application de DRI₁ ou de DRI₂ donne aux utilisateurs la possibilité de saisir les éléments de base de la requête.

4.2.1.1. Réaliser « Formuler une requête Par langage naturel » (DRI₁)

La DRI₁ est associée à la section <Démarrer, Formuler une requête, Par langage naturel > de la carte Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Formuler une requête* en utilisant la stratégie *Par langage naturel*.

La DRI₁ consiste à utiliser le langage naturel sous forme de mots clés pour capturer les éléments de but. Mais, sans identifier la nature de chaque élément (verbe, objet...). Elle concerne les utilisateurs qui ne connaissent ni le formalisme de but ni comment ils peuvent en servir. Ils préfèrent chercher les services intentionnels de la même façon qu'ils utilisent les moteurs de recherche sur internet (Google...). Ils saisissent leurs requêtes comme texte libre ou une combinaison de mots clés. Par exemple, « *traiter un patient* ».

Pour reformuler une requête de l'utilisateur, nous cherchons dans les mots de la requêtes si l'un parmi eux appartient à l'ontologie des produits ; s'il y a en un ou plusieurs, nous pouvons

proposer à l'utilisateur la reformulation à partir de ces produits, où nous pouvons extraire les verbes qui sont reliés avec ce produit et les produits adjacents.

En appliquant DRI_1 sur l'exemple « *traiter un patient* », le mot « *patient* » existe dans l'ontologie des produits (l'ontologie médicale). Donc, nous pouvons proposer à l'utilisateur toutes les formulations à partir de ce concept « *patient* ». Puisque le concept « *patient* » est relié à la classe de verbes « *interagir d'une certaine manière* » qui contient les verbes {« *traiter* », « *soigner* », « *examiner* »}, et il est relié à la classe de verbes « *obtenir quelque chose; entrer en possession de* » qui contient les verbes {« *accueillir* », « *recevoir* »} (la Figure 5.7).

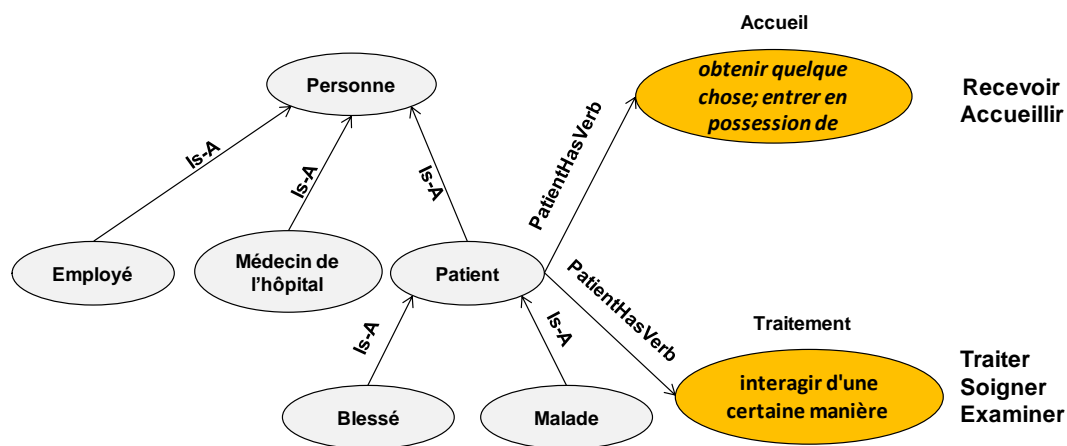


Figure 5.7. Le concept « Patient » dans l'ontologie des produits

Dans ce cas, nous aurons les premières reformulations suivantes :

Traiter un patient, soigner un patient, examiner un patient.

D'autres reformulations sont à proposer à l'utilisateur :

Traiter une personne, soigner une personne, examiner une personne, traiter un malade, soigner un malade, examiner un malade...

4.2.1.2. Réaliser « Formuler une requête Par Template » (DRI_2)

La DRI_2 est associée à la section <Démarrer, Formuler une requête, Par Template > de la carte Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Formuler une requête* en utilisant la stratégie *Par Template*.

La DRI_2 consiste à utiliser un template pour capturer les éléments du but (Figure 5.8). L'utilisateur saisit les éléments de but ou une partie de ces éléments constituant sa requête. Dans cette directive, les utilisateurs sont capables d'identifier les éléments de but (verbe, objet, ...), et ils sont capables de saisir ces éléments dans un template de but prédéfini pour cet objectif.

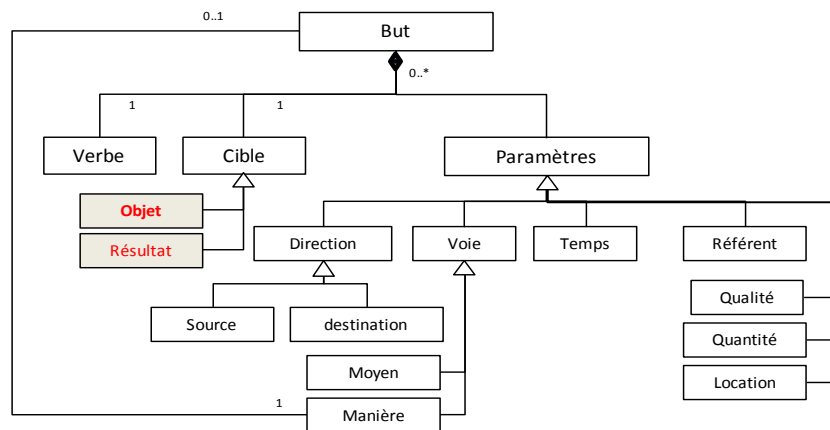


Figure 5.8. Structure d'un but

Cette directive a été introduite comme stratégie alternative de la première, et le template est basé initialement sur le modèle de but. La sortie est la requête de l'utilisateur structurée selon le modèle de but. Dans cette structuration, les éléments constituant le but sont identifiés.

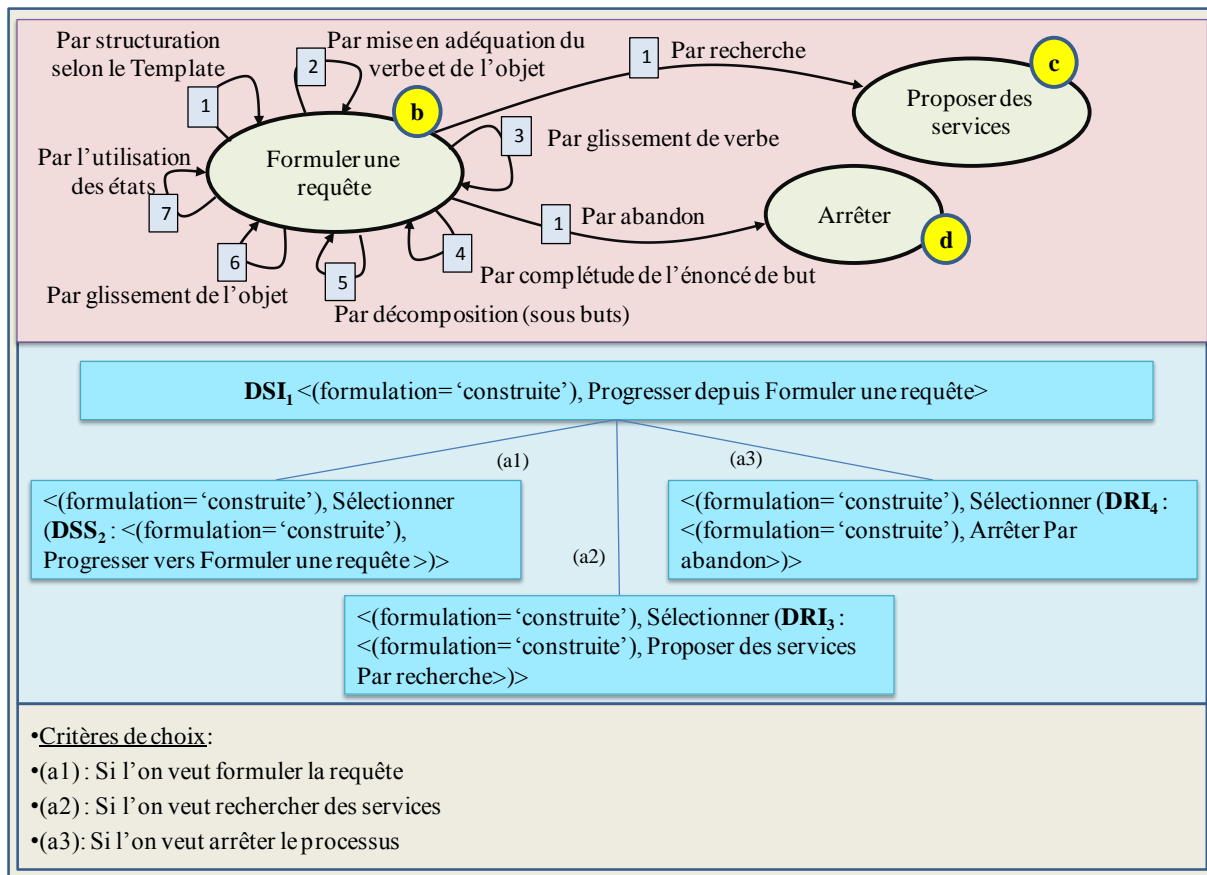
Par exemple, dans la requête « *recevoir un patient sur rendez-vous* », l'utilisateur détermine que « *recevoir* » est un verbe, « *patient* » est un objet et « *sur rendez-vous* » est une manière.

Le résultat est la requête suivante (ou le but) :

(*Recevoir*) *verbe* (*patient*) *produit* (*sur rendez-vous*) *manière*

4.2.2. Progresser depuis « Formuler une requête » (DSI_1)

La directive DSI_1 permet de progresser à partir de l'intention « *Formuler une requête* ». Le Map-Recherche indique qu'à partir de cette intention, il est possible de progresser en sélectionnant l'intention « *Proposer des services* », « *Formuler une requête* » ou « *Arrêter* ». Par conséquent, la DSI_1 est une directive de type choix composée de trois alternatives. Les trois alternatives sont des directives exécutables, chacune associée à une action (Figure 5.9). Elles proposent de sélectionner la directive de sélection de stratégie DSS_2 et les directives de réalisation d'intentions DRI_3 et DRI_4 .

Figure 5.9. La structure de DSI₁

La Directive DSS₂ (<(formulation= 'construite'), Progresser vers Formuler une requête >) guide le choix d'une stratégie de progression vers l'intention « Formuler une requête » (Figure 5.9-partie supérieure). Cette directive est sélectionnée suivant l'argument a1.

La Directive DRI₃ (<(formulation= 'construite'), Proposer des services Par recherche>) guide le choix d'une stratégie de réalisation de l'intention « Proposer des services », (Figure 5.8-partie supérieure). Cette directive est sélectionnée suivant l'argument a2.

La Directive DRI₄ (<(formulation= 'construite'), Arrêter Par abandon >) guide le choix d'une stratégie de réalisation de l'intention « Arrêter », (Figure 5.8 -partie supérieure). Cette directive est sélectionnée suivant l'argument a3.

Trois arguments (a1), (a2) et (a3) sont proposés pour guider le choix :

- (1) Si l'on veut formuler la requête
- (2) Si l'on veut rechercher des services
- (3) Si l'on veut arrêter le processus.

En effet, si l'utilisateur souhaite effectuer une formulation de sa requête, il peut décider de sélectionner la DSS₂, si l'utilisateur dispose de suffisamment de reformulations et décide de lancer la recherche, il peut choisir la DRI₃ et si l'utilisateur décide d'abandonner le processus, il pourra décider de sélectionner la DRI₄.

La DRI₃ et la DRI₄ sont présentées respectivement aux paragraphes 4.2.2.1 et 4.2.2.2. La DSS₂ est présentée dans le paragraphe 4.2.2.3.

4.2.2.1. Réaliser « Proposer des services Par recherche » (DRI₃)

La DRI₃ est associée à la section bc₁ du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Proposer des services* en utilisant la stratégie *Par recherche* (*faire coïncidence le verbe et l'objet, faire coïncidence le verbe ou faire coïncidence l'objet*).

Si l'utilisateur dispose de reformulations et s'il décide de lancer la recherche, en sélectionnant la directive DRI₃, l'utilisateur sélectionne cette directive pour rechercher des services intentionnels disponibles dans l'annuaire par (*faire coïncidence du verbe et de l'objet, faire coïncidence le verbe ou faire coïncidence l'objet*). Nous commençons par la recherche par *faire coïncidence le verbe et l'objet*, et nous utilisons le matching par le verbe ou par l'objet dans le cas où la première recherche n'a pas abouti. Cependant, le résultat de la recherche est une liste de services candidats qui correspondent à la requête reformulée par l'utilisateur. Nous prenons comme entrée à cette directive une reformulation donnée, nous transformons les éléments de cette reformulation au format d'un descripteur de service intentionnel, puis nous lançons la recherche. Nous pouvons utiliser les algorithmes standards d'un annuaire de services sémantiques. La sortie de cette stratégie est un ensemble d'ensembles de services, chaque sous-ensemble est la réponse à une reformulation. Le résultat total est normalement trié, parce que nous avons exécuté les reformulations par ordre.

Dans DRI₃, nous comparons la reformulation une par une avec les services disponibles dans l'annuaire, où nous cherchons pour chaque reformulation les services qui ont le même verbe et le même objet (la cible). Le résultat obtenu est trié par degré de similarité. Ce degré est calculé selon la distance dans l'ontologie *pOnto* entre l'objet utilisé dans la requête et l'objet du service trouvé, le poids du verbe n'a pas d'importance sur ce degré puisque tous les verbes de la même classe de sens de verbes sont équivalents. Par exemple, la distance entre « *blessé* » et « *personne* » est (2α), et la distance entre « *blessé* » et « *docteur* » est ($2\alpha\beta$). Ce qui signifie que « *blessé* » est plus proche de « *personne* » que de « *docteur* », et que le service intentionnel « *accueillir personne* » répond mieux que le service intentionnel « *recevoir employé* » à la requête « *recevoir blessé* », portant le service intentionnel « *recevoir employé* » a le même verbe « *recevoir* » que la requête, c'est parce que les deux verbes « *recevoir* » et « *accueillir* » sont équivalents.

Le résultat de la reformulation de « *accueillir patient* » est la liste triée :

- « *accueillir personne* » avec un degré de similarité α
- « *accueillir blessé* » avec un degré de similarité β
- « *accueillir employé* » avec un degré de similarité ($\alpha*\beta$)
- « *accueillir secrétaire* » avec un degré d'appariement ($\alpha*\beta^2$)
- ...

4.2.2.2. Réaliser « Arrêter Par abandon » (DRI₄)

La DRI₄ est associée à la section bc₁ du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Arrêter* en utilisant la stratégie *Par abandon*. Si l'utilisateur décide d'abandonner le processus de recherche.

4.2.2.3. Progresser vers « Formuler une requête » (DSS₂)

Le DSS₂ est de type choix. Elle permet de progresser à partir de l'intention « *Formuler une requête* » vers l'intention « *Formuler une requête* », c'est-à-dire qu'elle permet de boucler sur l'intention elle-même. Elle offre sept stratégies pour réaliser cela : *Par structuration selon le*

Template, Par mise en adéquation du verbe et de l'objet, Par glissement de verbe, Par complétude de l'énoncé de but, Par décomposition (sous buts), Par glissement de l'objet et Par l'utilisation des états. La Figure 5.10 montre la structure de cette DSS₂ ainsi que les arguments pour le choix d'une stratégie plutôt qu'une autre.

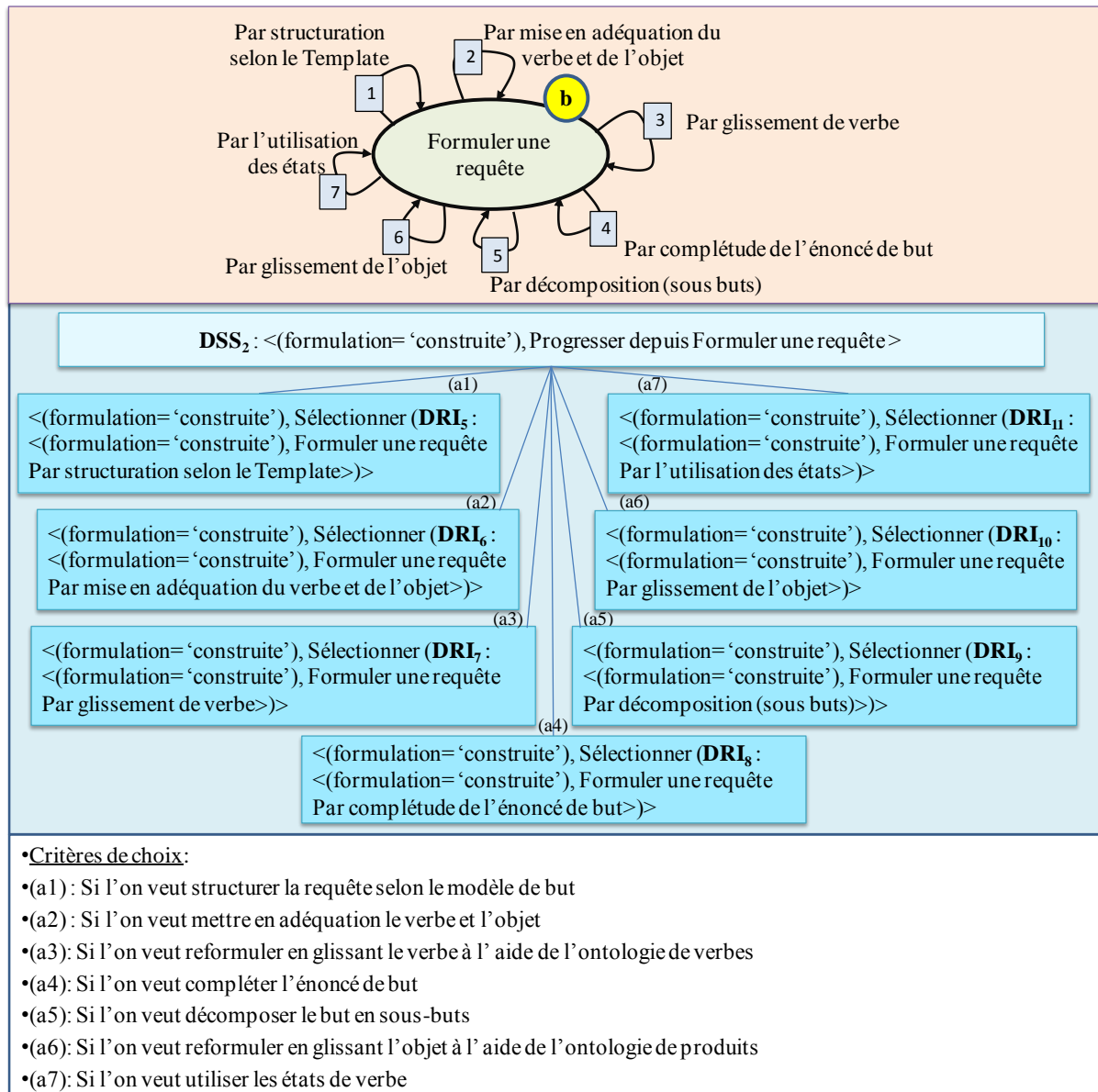


Figure 5.10. La structure de DSS₂

Dans la suite, nous détaillons les 7 DRI associées à DSS₂ qui ont été identifiées. Ces DRI sont :

- DRI₅ : <(formulation= 'construite'), Formuler une requête Par structuration selon le Template>
- DRI₆ : <(formulation= 'construite'), Formuler une requête Par mise en adéquation du verbe et de l'objet>
- DRI₇ : <(formulation= 'construite'), Formuler une requête Par glissement de verbe>

- DRI₈ : <(formulation= ‘construite’), Formuler une requête Par complétude de l’énoncé de but>
- DRI₉ : <(formulation= ‘construite’), Formuler une requête Par décomposition (sous buts)>
- DRI₁₀ : <(formulation= ‘construite’), Formuler une requête Par glissement de l’objet>
- DRI₁₁ : <(formulation= ‘construite’), Formuler une requête Par l’utilisation des états>

4.2.2.3.1. Réaliser « Formuler une requête Par structuration selon le template » (DRI₅)

La DRI₅ est associée à la section bb₁ du Map-Recherche (Figure 5.5). Elle vise à réaliser l’intention *Formuler une requête* en utilisant la stratégie *Par structuration selon le template*.

Cette directive consiste à structurer les éléments de la requête. Ceci concerne l’utilisateur qui a saisi la requête et n’était pas capable d’identifier les éléments de but (verbe, objet, ...). Cette directive a été introduite comme directive complémentaire à la première (DRI₁). Le résultat de cette directive est la requête de l’utilisateur structurée selon le modèle de but. Dans cette structuration, les éléments constituant le but sont identifiés.

Par exemple, dans la requête « *recevoir un patient sur rendez-vous* », l’utilisateur indique que « *recevoir* » est un verbe, « *patient* » est un objet et « *rendez-vous* » est une manière.

Le résultat est la requête suivante (ou le but) :

(*Recevoir*)_{verbe} (*patient*)_{objet} (*sur rendez-vous*)_{manière}

4.2.2.3.2. Réaliser « Formuler une requête Par mise en adéquation du verbe et de l’objet » (DRI₆)

La DRI₆ est associée à la section bb₂ du Map-Recherche (Figure 5.5). Elle vise à réaliser l’intention *Formuler une requête* en utilisant la stratégie *Par mise en adéquation du verbe et de l’objet*.

Cette directive aide l’utilisateur à s’assurer de la compatibilité entre le verbe et l’objet par examen du lien verbe-produit dans l’ontologie des produits. Nous prenons un objet, et nous cherchons s’il a une relation définie dans l’ontologie des produits avec une classe de sens du verbe :

- ✓ si une relation existe ; le produit est relié au verbe avec le lien « *produitHasVerbe* » et ils sont compatibles.
- ✓ sinon il n’y a pas de relation qui relie le produit avec la classe de sens du verbe ; on propose à l’utilisateur de modifier le verbe en introduisant les classes de sens de verbe qui sont reliées à ce produit.

A la Figure 5.11, nous présentons un exemple qui montre la relation entre le produit et le verbe en utilisant l’ontologie des produits *pOnto* (l’ontologie de domaine santé). Dans cette ontologie, nous avons ajouté un lien « *produitHasVerbe* » à chaque produit de l’ontologie. Ce lien concernant les verbes que ce produit accepte. Par exemple, le produit *patient* accepte les verbes appartenant à la classe de sens de verbes *d’accueil* comme « *recevoir, accueillir...* ». En examinant le lien que nous avons défini, nous pouvons vérifier la compatibilité entre le verbe et le produit.

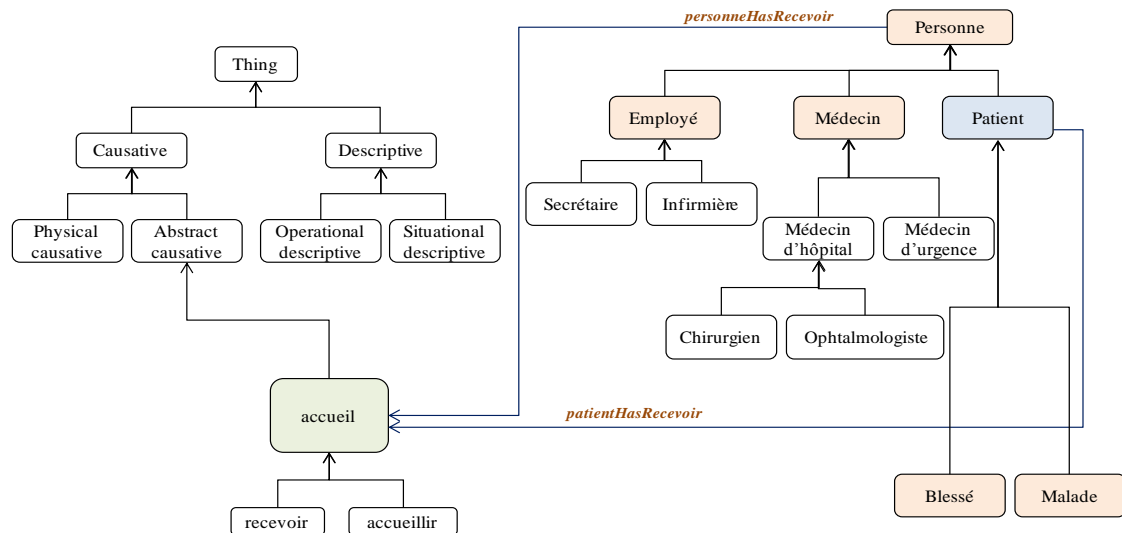


Figure 5.11. La vérification de la compatibilité entre verbe et objet

Dans l'exemple « *recevoir un patient* », on voit que le verbe et le produit sont compatibles : le objet « *patient* » accepte le verbe « *recevoir* » parce qu'il y a un lien « *patientHasRecevoir* ».

Au cas où le verbe n'appartient pas aux classes de sens de verbes que l'objet accepte, nous pouvons résoudre ce conflit en suggérant à l'utilisateur les verbes qu'il peut utiliser avec son produit. Par exemple, dans le but « *déclarer un patient* », on voit que le verbe et le objet ne sont compatibles : le objet « *patient* » n'accepte pas le verbe « *déclarer* » parce qu'il n'y a pas de lien entre le produit et la classe de sens de verbe « *proclamer un soutien* ». Nous suggérons à l'utilisateur d'utiliser les verbes appartenant aux classes de sens de verbe liés à ce produit.

4.2.2.3.3. Réaliser « *Formuler une requête Par glissement de verbe* » (DRI₇)

La DRI₇ est associée à la section bb₃ du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Formuler une requête* en utilisant la stratégie *Par glissement de verbe*.

Le glissement de verbe dans la requête consiste à remplacer le verbe. Le remplacement se fait par un autre verbe plus fréquent appartenant à la même classe de sens de verbe. Le verbe « *recevoir* » appartient à la classe de sens de « *verbes d'accueil* » qui contient les verbes « *recevoir* », « *accueillir* »... Une classe de sens de verbe est une sous-classe des seize classes rencontrées dans l'ontologie des verbes [Urrego, 2005].

Cette directive s'assure aussi du sens de verbe, parce qu'un verbe normalement appartient à une ou plusieurs classes de sens de verbes. Par conséquent, elle aide l'utilisateur dans la précision du sens de verbe, elle donne plus d'information sur la classe de sens de verbe et des exemples d'utilisation. Cette directive a pour but de donner à l'utilisateur la possibilité de choisir d'autres verbes ou de l'aider tout simplement à modifier le verbe s'il souhaite pour améliorer l'expressivité de son but.

Prenons l'exemple du but « *recevoir un patient sur rendez-vous* », le verbe « *recevoir* » appartient à plusieurs classes de sens de verbes « *verbes d'accueil* », « *obtenir quelque chose; entrer en possession de* » ... Mais, le produit « *patient* » a un lien seulement avec « *verbes d'accueil* », qui contient le verbe « *accueillir* », Figure 5.12.

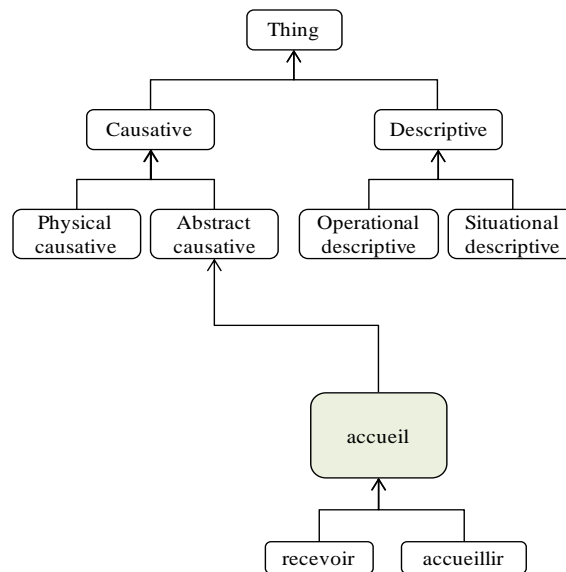


Figure 5.12. Glissement de verbe dans l'ontologie des verbes

Dans ce cas, l'utilisateur a le choix d'utiliser le verbe qui convient ; soit il garde le verbe ou bien il le remplace par un autre plus fréquent de la même classe de sens de verbe.

4.2.2.3.4. Réaliser « Formuler une requête Par complétude de l'énoncé de but » (DRI₈)

La DRI₈ est associée à la section bb₄ du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Formuler une requête* en utilisant la stratégie *Par complétude de l'énoncé de but*.

La complétude consiste à prévenir l'utilisateur aux autres paramètres constituant les éléments de but selon la signature de verbe [Urrego, 2005]. La signature de verbe dépend des seize classes de verbes dans l'ontologie *vOnto*, Figure 5.13. Par exemple, le verbe « *accueillir* » est un verbe *causatif intellectuel de service*. D'un côté, les verbes de *service* peuvent utiliser tous les paramètres (tous les paramètres sont facultatifs). De l'autre côté, parce que le verbe « *recevoir* » est causatif, il affecte la relation avec le produit « *patient* » et accepte surtout, le paramètre *voie* (*manière/moyen*) qui indique comment cette relation a été affectée.

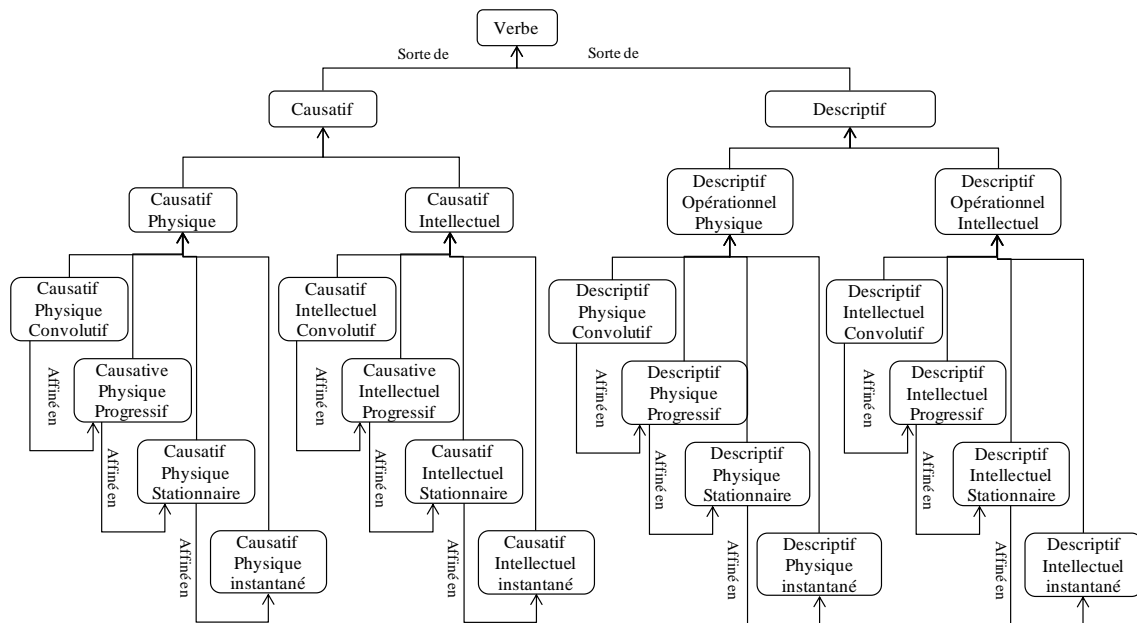


Figure 5.13. L'ontologie des verbes

Dans l'exemple « *recevoir un patient* », la nouvelle reformulation est la suivante :

(Recevoir) *verbe* (*patient*) *objet* (*sur rendez-vous*) *manière*

4.2.2.3.5. Réaliser « *Formuler une requête Par décomposition (sous buts)* » (DRI₉)

La DRI₉ est associée à la section bb₅ du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Formuler une requête* en utilisant la stratégie *Par décomposition (sous buts)*.

La décomposition est seulement dans le cas où la manière se présente dans le but. La manière identifie une approche selon laquelle le but peut être satisfait. Dans, *(Annuler)* *verbe* (*une réservation*) *résultat* (*par expiration du délai d'attente*) *manière*. Par expiration du délai d'attente est la manière pour satisfaire le but Annuler une réservation.

Dans la manière on précise la façon dont le but va chercher à être atteint. Ce paramètre apporte de la précision dans la formulation du but. On comprend qu'il puisse être important lorsque le but qualifie un service de rendre la manière explicite car la valeur de ce paramètre renseigne sur le processus d'atteinte du but qui sera exécuté par les services logiciels sous jacents au service intentionnel [Kaabi, 2007].

La Figure 5.8 montre que seule la manière est un paramètre complexe puisqu'une manière peut être un but. Par exemple,

(Améliorer) *verbe* (*les services offerts aux clients d'une agence de voyages*) *cible* (*en mettant en place un système de réservation sur Internet*) *manière* est un but dont la manière 'en mettant en place un système électronique de réservation' est un but : *(Mettre en place)* *verbe* un (*système de réservation*) *résultat* (*sur Internet*) *moyen*.

On voit que cette possibilité est utile pour décrire un but en le décomposant en deux buts. Dans l'exemple précédent, le but *Améliorer les services offerts aux clients d'une agence de voyages* et le but *Mettre en place système de réservation sur Internet*.

4.2.2.3.6. Réaliser « Formuler une requête Par glissement de l'objet » (DRI₁₀)

La DRI₁₀ est associée à la section bb₆ du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Formuler une requête* en utilisant la stratégie *Par glissement de l'objet*.

Cette directive consiste à chercher les produits adjacents dans l'ontologie des produits. Dans une ontologie de domaine, on trouve plusieurs types de liens entre les objets de l'ontologie. Mais, nous nous intéressons seulement aux liens de généralisation « *isA* » et de spécialisation « *has* », parce que ces liens sont équivalents à la synonymie/hyperonymie/hyponymie dans le dictionnaire, et donnent des sens proches du celui du concept recherché. Tandis que, les autres liens donnent autres informations qui sont sémantiquement moins importantes.

Prenons l'exemple du produit « *patient* », on trouve que « *patient* » est une instanciación de « *personne* », une généralisation de « *malade* » et « *blessé* » ou proche de « *docteur* », « *infirmière* », « *employé* »... L'objectif est de donner à l'utilisateur la possibilité de choisir le *produit* approprié ou de lui aider à le modifier s'il souhaite pour améliorer l'expressivité du but, Figure 5.14.

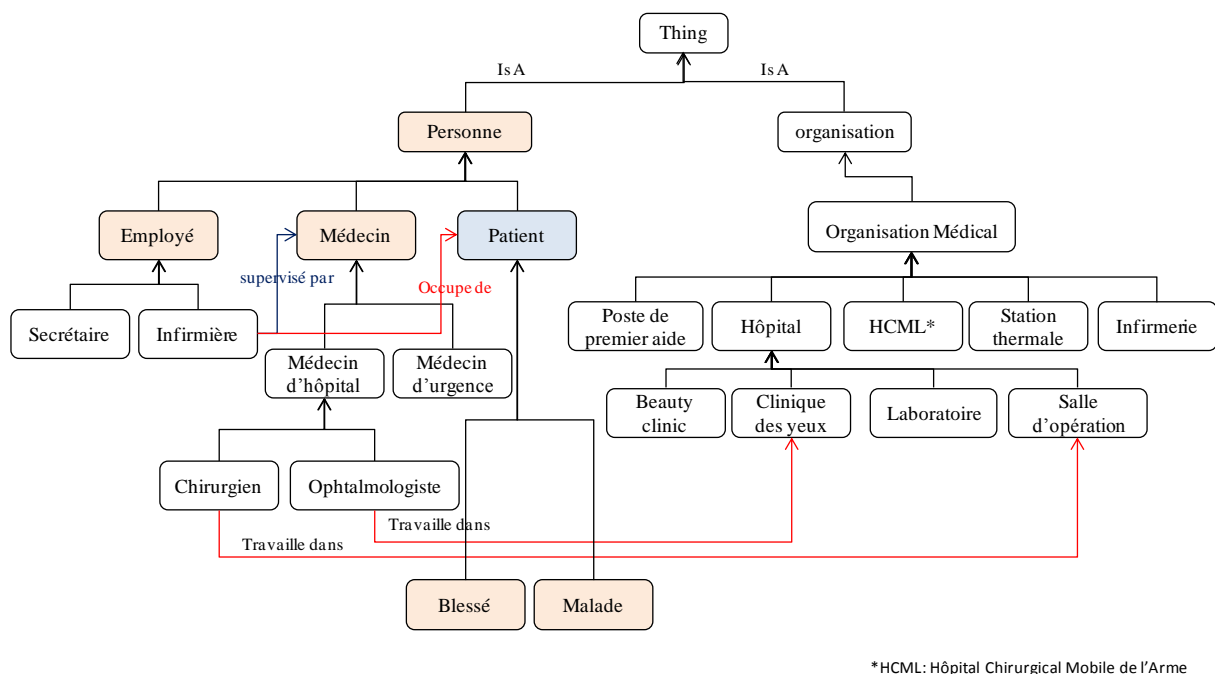


Figure 5.14. Glissement de l'objet dans l'ontologie des produits

Le résultat de cette directive appliquée sur la requête « *recevoir un patient* » est la reformulation suivante :

Recevoir un patient, recevoir une personne, recevoir un malade, recevoir un blessé, recevoir un docteur, recevoir un employé...

4.2.2.3.7. Réaliser « Formuler une requête Par l'utilisation des états » (DRI₁₁)

La DRI₁₁ est associée à la section bb₇ du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Formuler une requête* en utilisant la stratégie *Par l'utilisation des états*.

Cette directive a été utilisée dans le MAP-Publication, et elle a été expliquée dans le chapitre 5. Par exemple, un verbe de *service* énonce un développement par étapes. Les verbes de cette

catégorie considèrent le changement de *situation*. Autrement dit, les verbes de *service* acceptent les paramètres *situation* (*situation initiale* ou *situation finale* ou les deux). Pour les verbes de *service* les situations décrites par les paramètres *état initial* et *état final* représentent des *processus*, c'est-à-dire que ces *situations* sont énoncées par des verbes de *processus* et ont aussi un développement par étapes.

Les *situations* énoncées par les verbes de service *causatifs* sont des *situations* de leur paramètre *objet*, alors que les *situations* traitées par les verbes de service *descriptifs* sont des *situations* des *relations avec l'objet*.

On reprend l'exemple du verbe *accueillir*. Ce verbe est du type *causatif physique* et appartient à la catégorie de verbes de *service* qui affecte la relation avec le objet sans le faire évoluer, les buts exprimés par des verbes de *service* sont nommés *convolutifs* pour indiquer que leur accomplissement est une convolution de processus, c'est-à-dire que les buts *convolutifs* représentent un *processus* de *processus*. Donc, les états de la relation *accueillir patient* changent :

accueillir_Patient.état_initial = 'pas de registre' OU

accueillir_Patient.état_final = 'enregistré' OU

accueillir_Patient.état_final = 'reporté'.

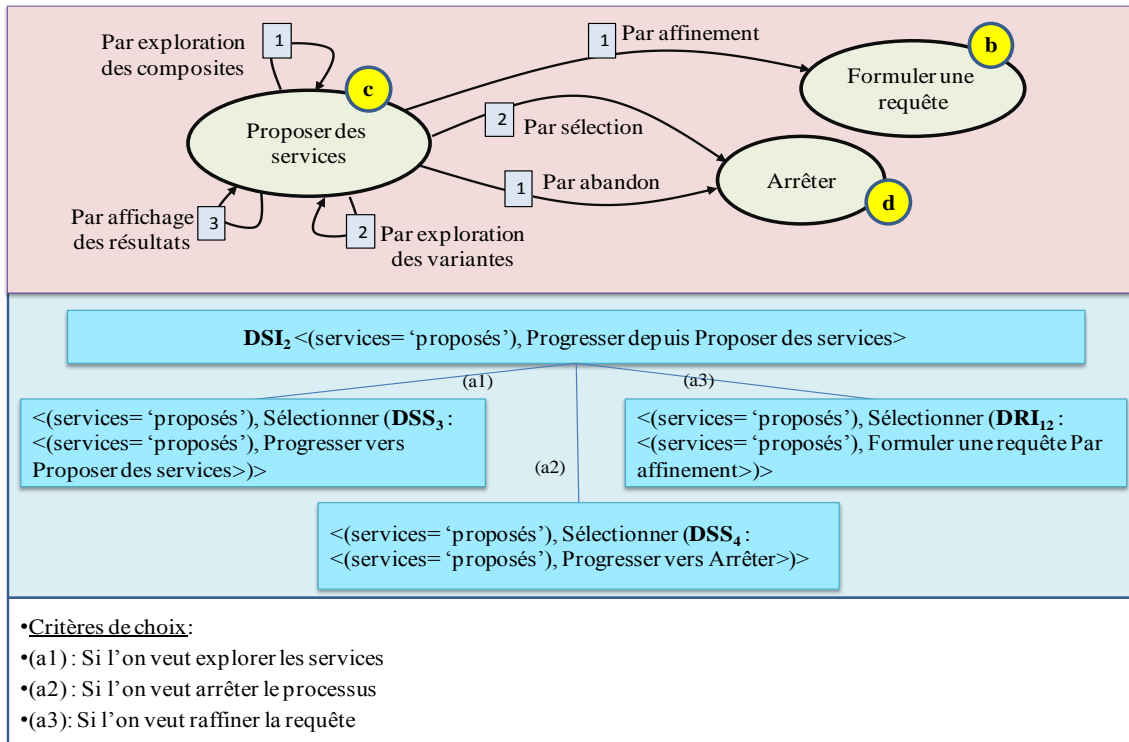
En effet, ce processus fait évoluer la *relation* avec le produit « *patient* » de l'état initial « *le manque de registre* » jusqu'à l'état final « *l'accomplissement d'enregistrement* ».

Les verbes de *processus* comme les verbes de *service* indiquent un développement par étapes impliquant le passage de *l'objet* d'un *état initial* à un *état final* différent.

Tandis que les verbes d'*activité* et les verbes d'*action* ne considèrent pas le changement d'*état*.

4.2.3. Progresser depuis « Proposer des services » (DSI₂)

La directive DSI₂ permet de progresser à partir de l'intention « *Proposer des services* ». Le Map-Recherche indique qu'à partir de cette intention, il est possible de progresser en sélectionnant l'intention « *Proposer des services* », « *Formuler une requête* » ou « *Arrêter* ». Par conséquent, la DSI₂ est une directive de type choix composée de trois alternatives. Les trois alternatives sont des directives exécutables, chacune associée à une action (Figure 5.15). Elles proposent de sélectionner la directive de sélection de stratégie DSS₃ et DSS₄ et les directives de réalisation d'intentions DRI₁₂.

Figure 5.15. La structure de DSI₂

La Directive DSS₃ (<(services= 'proposés'), Progresser vers Proposer des services >) guide le choix d'une stratégie de progression vers l'intention « Proposer des services » (Figure 5.16-partie supérieure). Cette directive est sélectionnée suivant l'argument a1.

La Directive DSS₄ (<(services= 'proposés'), Progresser vers Arrêter >) guide le choix d'une stratégie de progression vers l'intention « Arrêter » (Figure 5.15-partie supérieure). Cette directive est sélectionnée suivant l'argument a2.

La Directive DRI₁₂ (<(services= 'proposés'), Formuler une requête Par affinement >) guide le choix d'une stratégie de réalisation de l'intention « Formuler une requête », (Figure 5.15-partie supérieure). Cette directive est sélectionnée suivant l'argument a3.

Trois arguments (a1), (a2) et (a3) sont proposés pour guider le choix :

- (a1) : Si l'on veut proposer des services
- (a2) : Si l'on veut arrêter le processus
- (a3) : Si l'on veut reformuler la requête.

En effet, si l'utilisateur souhaite proposer des services par exploration des services disponibles dans l'annuaire, il peut décider de sélectionner la DSS₃. Si l'utilisateur ne dispose pas de suffisamment de proposition et il veut reformuler sa requête, il peut choisir la DRI₁₂. Si l'utilisateur décide de sélectionner un service ou d'abandonner le processus, il pourra décider de sélectionner la DSS₄.

La DRI₁₂ est présentée dans le paragraphe 4.2.3.1. La DSS₃ et la DSS₄ sont présentées respectivement aux paragraphes 4.2.3.2 et 4.2.3.3.

4.2.3.1. Réaliser « Proposer des services Par affinement » (DRI₁₂)

La DRI₁₂ est associée à la section cb₁ du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Formuler une requête* en utilisant la stratégie *Par affinement*.

L'affinement de la requête consiste à modifier la requête originale de l'utilisateur. Cette directive est utilisée dans le cas où la recherche de services a rendu un résultat non satisfaisant et que les services trouvés sont loin des besoins de l'utilisateur. Dans ce cas, l'utilisateur modifie les éléments de but ou une partie de ces éléments constituant sa requête. La modification concerne le verbe ou le objet ou les paramètres ou tous en même temps. Le résultat de cette directive est une requête modifiée qui respecte le modèle de but.

Après avoir modifié la requête, l'utilisateur peut utiliser le résultat de la modification comme entrée pour une nouvelle reformulation.

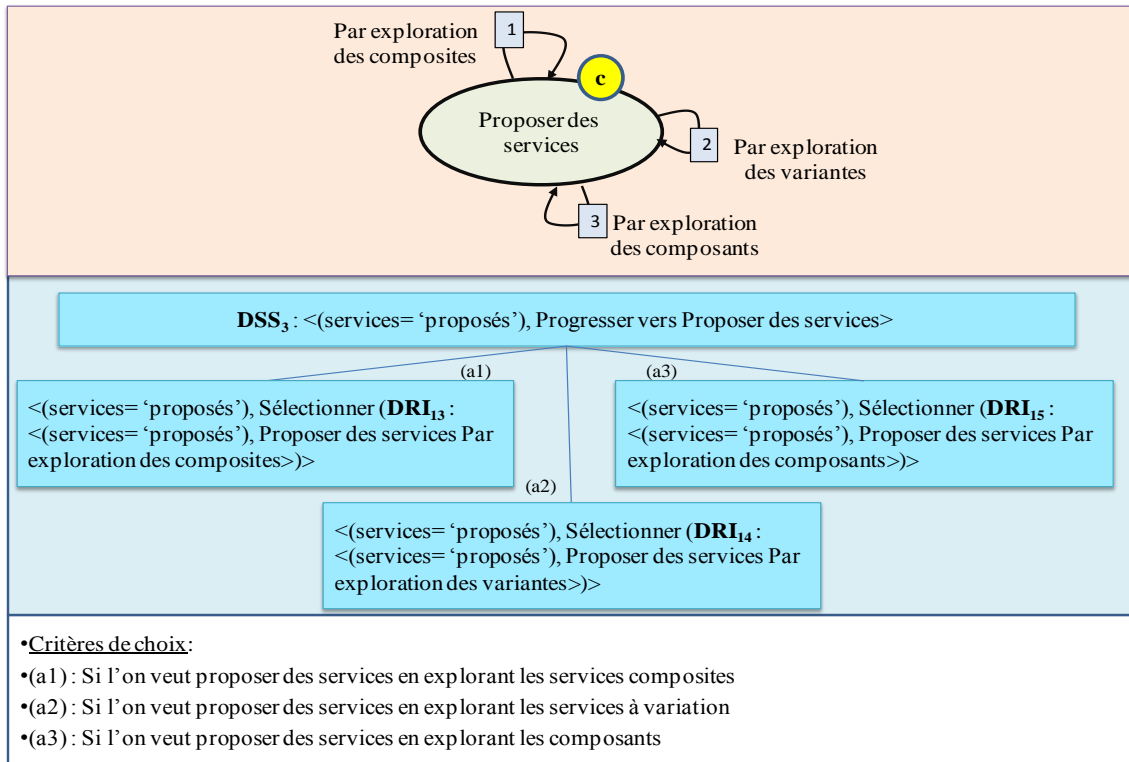
4.2.3.2. Progresser vers « Proposer des services » (DSS₃)

Le DSS₃ est de type choix. Elle permet de progresser à partir de l'intention « *Proposer des services* » vers l'intention « *Proposer des services* », c'est-à-dire qu'elle permet de boucler sur l'intention elle-même. Elle offre trois stratégies pour réaliser cela : *Par exploration des composites*, *Par exploration des variantes* et *Par exploration des composants*. La Figure 5.16 montre la structure de cette DSS₃ ainsi que les arguments pour le choix d'une stratégie plutôt qu'une autre.

Après la proposition des services répondant à une requête, l'utilisateur peut explorer les services agrégats. Un *service agrégat* est composé d'autres services qui peuvent être atomiques ou eux-mêmes des services agrégats. Deux types de services agrégats, les *services à variation* et les *services composites*. Les premiers établissent des liens OU entre services composants et offrent des choix dans la manière d'atteindre le but du service ; les seconds établissent des liens ET entre services composants.

Dans la suite, nous détaillons les 3 DRI associées à DSS₃ qui ont été identifiées. Ces DRI sont :

- DRI₁₃: <(services= 'proposés'), Proposer des services Par exploration des composites>
- DRI₁₄: <(services= 'proposés'), Proposer des services Par exploration des variantes >
- DRI₁₅ : <(services= 'proposés'), Proposer des services Par exploration des composants>

Figure 5.16. La structure de DSS₃

Dans la section 4.2.3.2.1, nous explorons les services composites, dans la section 4.2.3.2.3, nous traitons les services composants, tandis que dans la section 4.2.3.2.2, nous traitons les services à variation.

4.2.3.2.1. Réaliser « Proposer des services Par exploration des composites » (DRI₁₃)

La DRI₁₃ est associée à la section cc₁ du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Proposer des services* en utilisant la stratégie *Par exploration des composites*.

Pour illustrer cette directive, nous prenons l'exemple des réservations de voitures, le service de confirmation d'une réservation *S Réserver une voiture* est composé de deux services : *S Effectuer réservation* et *S Payer réservation*. Ceci se traduit par la composition séquentielle de services suivante : *S Réserver une voiture* = • (*S Effectuer réservation*, *S Payer réservation*).

Dans un deuxième exemple des réservations hôtelières (la Figure 5.17), le service d'annulation d'une réservation *S Annuler réservation* est composé de deux services : *S Effectuer réservation* et *S Payer réservation*. Ceci se traduit par la composition séquentielle de services suivante : *S Annuler réservation* = • (*S Effectuer réservation*, *S Payer réservation*).

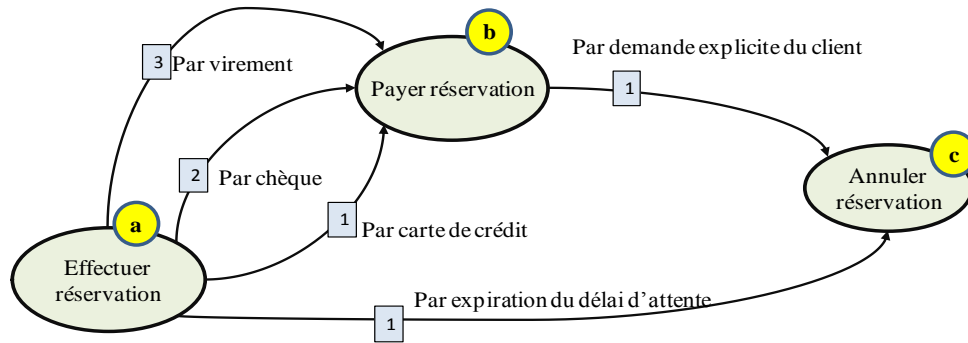


Figure 5.17. Le service à variation $S_{\text{Payer réservation}}$

Un utilisateur qui recherche un service pour payer une réservation, trouve le service $S_{\text{Payer réservation}}$. Ce service est un composant dans deux services composites : $S_{\text{Réserver une voiture}}$ et $S_{\text{Annuler réservation}}$. La directive nous permet d'explorer les deux services composites dont le service recherché est un composant.

4.2.3.2.2. Réaliser « Proposer des services Par exploration des variantes » (DRI_{14})

La DRI_{14} est associée à la section cc_2 du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Proposer des services* en utilisant la stratégie *Par exploration des variantes*.

Un service à variation correspond à une situation qui nécessite l'exploration de différentes possibilités alternatives car ce sont des situations dans lesquelles il existe différentes façons d'atteindre un but. Un service à choix alternatif exprime un choix d'alternatives dans la manière de réaliser le but du service agrégat en offrant des services composants qui sont mutuellement exclusifs. Chaque service représente une manière différente pour satisfaire le but de l'agrégat. Ainsi, un service à choix alternatif regroupe plusieurs services mutuellement exclusifs et construit ainsi un nouveau service de même niveau d'abstraction mais de granularité plus élevée [Kaabi, 2007]. Tandis qu'un service à choix multiple offre un choix non exclusif de réalisation du but du service agrégat en groupant plusieurs services parmi lesquels au moins mais éventuellement plusieurs pourront être choisis.

Dans la Figure 5.18 des réservations hôtelières, un utilisateur qui recherche à payer une réservation, trouve le service $S_{\text{Payer réservation}}$ qui peut être satisfait de différentes façons : le paiement de la réservation peut se faire par exemple *par carte de crédit*, *par chèque* ou *par virement*. Cela conduit à l'exploration de trois services $S_{\text{Payer réservation par carte de crédit}}$, $S_{\text{Payer réservation par chèque}}$ et $S_{\text{Payer réservation par virement}}$. Ces services composent le service à variation à choix alternatif et ils sont proposés à l'utilisateur par cette directive.

Dans le même exemple des réservations hôtelières, le service $S_{\text{Choisir vol}}$ est à choix multiple. Il existe en effet plusieurs manières non exclusives de satisfaire le but *Choisir vol*. On peut utiliser différents critères tels que le prix, la compagnie aérienne ou les horaires pour effectuer le choix. Il est composé de $S_{\text{Choisir un vol par compagnie aérienne}}$, $S_{\text{Choisir un vol par prix}}$ et $S_{\text{Choisir un vol par horaire}}$. Pour un utilisateur qui recherche un service pour choisir un vol, cette directive lui permet d'explorer les variantes de ce service.

4.2.3.2.3. Réaliser « Proposer des services Par exploration des composants » (DRI₁₅)

La DRI₁₅ est associée à la section cc₃ du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Proposer des services* en utilisant la stratégie *Par exploration des composants*.

Dans cette directive, nous explorons les services composants d'un service composite. Pour illustrer cette directive, nous prenons le même exemple des réservations hôtelières (la Figure 5.18), le service d'annulation d'une réservation *S Annuler réservation* est composé de deux services : *S Effectuer réservation* et *S Payer réservation*.

Un utilisateur qui recherche un service pour annuler une réservation, trouve le service *S Annuler réservation*. Ce service est un service composite de deux services composants : *S Effectuer réservation* et *S Payer réservation*. La directive nous permet d'explorer les deux services composants.

4.2.3.3. Progresser vers « Arrêter » (DSS₄)

Le DSS₄ est de type choix. Elle permet de progresser à partir de l'intention « Proposer des services » vers l'intention « Arrêter ». Elle offre deux stratégies pour réaliser cela : *Par sélection* et *Par abandon*. La Figure 5.18 montre la structure de cette DSS₄ ainsi que les arguments pour le choix d'une stratégie plutôt que l'autre.

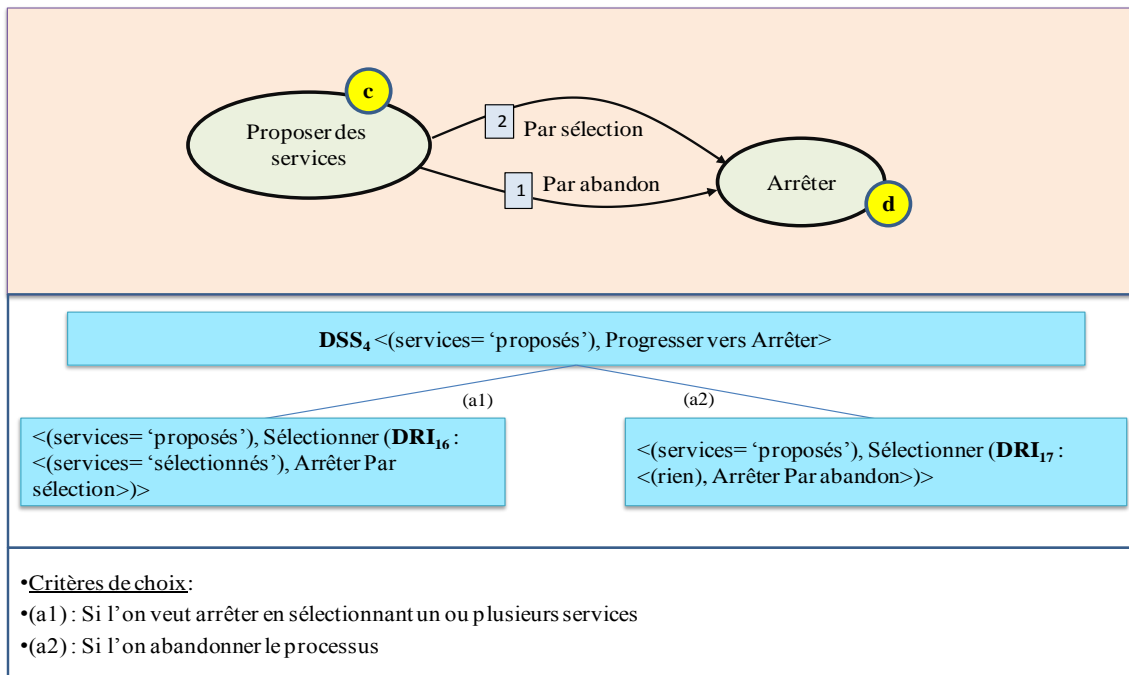


Figure 5.18. La structure de DSS₄

Dans la suite, nous détaillons les 2 DRI associées à DSS₄ qui ont été identifiées. Ces DRI sont :

- DRI₁₆ : <(services= 'Sélectionnés'), Arrêter Par sélection>
- DRI₁₇ : <(services= 'proposés'), Arrêter Par abandon>

4.2.3.3.1. Réaliser « Arrêter Par sélection » (DRI_{16})

La DRI_{16} est associée à la section cd_1 du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Arrêter* en utilisant la stratégie *Par sélection*.

Après avoir affiché le résultat, l'utilisateur a le choix de sélectionner le/les service(s) qu'il trouve approprié(s) à ces besoins.

4.2.3.3.2. Réaliser « Arrêter Par abandon » (DRI_{17})

La DRI_{17} est associée à la section cd_2 du Map-Recherche (Figure 5.5). Elle vise à réaliser l'intention *Arrêter* en utilisant la stratégie *Par abandon*.

5. SCENARIO D'UTILISATION

Cette section présente un exemple d'utilisation de la fonctionnalité de la reformulation de requêtes dans $\mathcal{P}\mathcal{A}\mathcal{S}\mathcal{I}\mathcal{S}$. Il s'agit de montrer comment les éléments présentés dans la section précédente sont utilisés dans le cadre d'un scénario particulier.

5.1. Principe général

$\mathcal{P}\mathcal{A}\mathcal{S}\mathcal{I}\mathcal{S}$ est principalement utilisé pour aider les utilisateurs dans leurs tâches de recherche de services. Ils utilisent donc $\mathcal{P}\mathcal{A}\mathcal{S}\mathcal{I}\mathcal{S}$ pour rechercher des services qui vont les aider à résoudre des problèmes de métier. Une interaction entre un utilisateur et $\mathcal{P}\mathcal{A}\mathcal{S}\mathcal{I}\mathcal{S}$ commence lorsqu'un utilisateur exprime un besoin sous la forme d'une requête.

La formulation d'une requête effectue une analyse de cette requête en recherchant des reformulations avec des termes définis dans les ontologies de $\mathcal{P}\mathcal{A}\mathcal{S}\mathcal{I}\mathcal{S}$. Ces termes faciliteront la recherche de services potentiels pour la réalisation de la requête.

Ensuite, la recherche de services met en correspondance les termes de chaque reformulation de la requête avec les éléments des services disponibles pour trouver des services qui répondent au besoin exprimé par l'utilisateur.

Enfin, les services trouvés à l'issue de la recherche sont proposés à l'utilisateur qui les explore et sélectionne les mieux adaptés à ses besoins.

Dans la section suivante, nous détaillons un exemple de requête exprimée par un utilisateur.

5.2. Présentation de l'exemple

On illustre l'utilisation de $\mathcal{P}\mathcal{A}\mathcal{S}\mathcal{I}\mathcal{S}$ par un scénario dans lequel un utilisateur souhaite payer la location d'une chambre d'hôtel, et il cherche un service intentionnel qui l'aide à faire ceci.

5.3. Déroulement du scénario

Le scénario commence avec l'expression de la requête « *Payer une chambre d'hôtel* ». La reformulation de ce but est présentée dans la Figure 5.19 en utilisant les directives de Map-Recherche.

La requête et ses formulations	Directive utilisée
(Payer) <i>verbe</i> (chambre d'hôtel) <i>objet</i>	DRI_2

(Payer) verbe (chambre d'hôtel) objet	DRI ₆
(Payer) verbe (chambre d'hôtel) objet (Régler) verbe (chambre d'hôtel) objet (Acquitter) verbe (chambre d'hôtel) objet (Verser) verbe (chambre d'hôtel) objet (Rembourser) verbe (chambre d'hôtel) objet	DRI ₇
(Payer) verbe (chambre d'hôtel) objet (Payer) verbe (hôtel) objet (Payer) verbe (réservation) objet (Payer) verbe (chambre simple) objet (Payer) verbe (chambre double) objet (Payer) verbe (salle de congrès) objet (Payer) verbe (appartement) objet (Payer) verbe (centre de jeunesse) objet (Payer) verbe (château) objet	DRI ₁₀

Figure 5.19. Formulation de la requête en utilisant les directives

Conformément aux directives du Map-Recherche (Figure 5.5 et tableau 6.1), le scénario se déroule de la façon suivante :

- A partir de la requête émise sous la forme du but « *Payer une chambre d'hôtel* », la directive DRI₂ consiste à structurer les éléments de la requête selon le modèle de but (verbe, objet, paramètres) et des termes définis dans les ontologies de *PASIS*.
- Nous pouvons appliquer la directive DRI₃ (*Proposer des services Par recherche*), elle exploite la base de services pour découvrir s'il existe des services possibles qui permettent de résoudre la requête formulée par l'utilisateur. Ces services doivent avoir le même verbe et le même objet. Le résultat obtenu dans ce cas est vide.
- Si on ne trouve pas de services qui ont le même verbe et le même objet, la directive DRI₁₃ permet de revenir en arrière (*formulation de requête*) afin de reformuler la requête.
- L'utilisation de la directive DRI₆ nous permet de s'assurer que le verbe de l'objet de la requête sont compatibles, cette vérification est basée sur les liens définis dans l'ontologie de produits avec l'ontologie de verbe. Dans notre exemple, le produit « *chambre d'hôtel* » accepte le verbe « *payer* » par l'existence d'un lien entre ce produit et la classe de sens de verbe « *donner de l'argent* (généralement en échange de biens ou services) » dont le verbe « *payer* » appartient.

- La directive DRI_7 permet de faire glisser le verbe. Dans l'ontologie des verbes, on cherche les verbes de la même classe de sens de verbe qui peuvent remplacer le verbe initial « *Payer* » qui appartient à la classe de sens de « *donner de l'argent en échange de biens ou services* » qui contient les verbes « *régler* », « *acquitter* », « *verser* » et « *rembourser* ».
- Dans l'ontologie des produit (dans notre exemple, c'est une ontologie de domaine touristique), on trouve que « *Chambre d'hôtel* » est une instanciation de « *hôtel* » qui est une instanciation de « *réservation* », une généralisation de « *simple* », « *double* », « *sale de congrès* »... ou proche de « *appartement* », « *centre de jeunesse* », « *château* »... Ces produits adjacents sont obtenus en appliquant la directive DRI_{10} et donnent à l'utilisateur la possibilité de faire glisser l'*objet* afin d'avoir d'autres reformulations de la requête. Nous pouvons classer le résultat à l'aide de l'algorithme de 6.1.
- Nous utilisons la reformulation (*Payer*)_{verbe} (*réservation*)_{objet} pour la recherche de services intentionnels en utilisant la directive DRI_3 .

À l'issue de la recherche, les différents services trouvés et leurs degrés de similarité avec la requête initiale de l'utilisateur, sont proposés à l'utilisateur dans la Figure 5.20.

Le service proposé	Directive
S <i>Payer une réservation</i>	DRI_3
S <i>Réserver une voiture</i> S <i>Annuler réservation.</i>	$DRI_{12.1}$
S <i>Payer une réservation par carte bancaire</i> S <i>Payer une réservation par chèque</i> S <i>Payer une réservation par virement</i>	$DRI_{12.2}$

Figure 5.20. Proposition des services en utilisant les directives

- Pour la valeur de $\alpha = 0.9$ et $\beta = 0.8$, le service dont le but est « *payer une réservation* », a le degré de similarité de 64%. En effet, le verbe « *payer* » est le même (similaire à 100%) et l'objet « *chambre d'hôtel* » est une spécialisation de « *réservation* » (similaire à 64 %). Par conséquent, le degré de similarité est égal à 66%, sachant que le verbe et l'objet font 80% du degré de similarité final.
- Le service S *Payer réservation* est un composant dans deux services composites : S *Réserver une voiture* et S *Annuler réservation*. La directive $DRI_{12.1}$ nous permet d'explorer les deux services composites dont le service recherché est un composant.
- En appliquant la directive $DRI_{12.2}$, nous obtenons les services « S *Payer une réservation par carte bancaire* », « S *Payer une réservation par chèque* » et « S *Payer une réservation par virement* » qui sont les composants du service à choix alternatif « S *Payer une réservation* ».
- L'utilisateur choisit le service qui réalise sa requête initiale.

Ce scénario décrit un exemple d'utilisation de l'approche *PASIS* qui permet d'obtenir plusieurs services intentionnels candidats avec degrés d'appariement différents pour la réalisation de la requête « *payer une chambre d'hôtel* ».

6. CONCLUSION

Les besoins des utilisateurs représentés par un but, en utilisant un formalisme linguistique, constituent dans notre travail la base pour un traitement systématique de leurs besoins à la recherche des services intentionnels.

Le problème traité dans ce chapitre concerne le guidage des utilisateurs dans leurs recherches de services et leurs formulations de requêtes. Pour cette raison, Nous avons proposé un processus de reformulation des requêtes des utilisateurs et de recherche de services. Ce processus aide l'utilisateur à reformuler sa requête en exploitant les ontologies, il guide l'utilisateur dans la reformulation et l'exploration du résultat. Nous avons défini un degré d'appariement à partir des relations dans les ontologies, ce degré mesure la distance entre la requête initiale et chaque reformulation. Finalement, nous avons illustré avec un exemple le déroulement du processus.

CHAPITRE 6

CAS D'APPLICATION

1. INTRODUCTION

Ce chapitre est consacré à une étude de cas qui illustre l'approche proposée dans cette thèse. Dans ce chapitre, nous illustrons le déroulement du processus de publication, et montrons comment il aide et assiste le fournisseur dans la publication des descriptions de services. Nous présentons aussi le déroulement du processus de guidage de l'utilisateur dans la formulation de requêtes et l'exploration des services à partir de sa requête. Nous montrons comment sélectionner les services les plus proches de besoins de l'utilisateur, et comment le guidage que propose le processus de reformulation de requête assiste l'utilisateur dans la navigation de l'espace sémantique associé à sa requête initiale à l'exploration des services intentionnels disponibles. Finalement, nous présentons un prototype en cours de développement de l'approche *PASIS*.

Ce chapitre est organisé comme suit : la section 2 présente le cas d'étude « *e-clinique* ». La section 3 présente le déroulement du processus de la publication. Ensuite, la section 4 le déroulement du processus de la recherche. Dans la section 5, nous discutons les hypothèses proposées au début de ce travail. Dans la section 6, nous illustrons le prototype par différentes captures d'écran. Finalement, nous concluons le chapitre par la section 7.

2. INTRODUCTION DU CAS D'ETUDE

Ce paragraphe présente l'étude de cas dans la publication des services intentionnels et introduit la description des services de l'application « *e-clinique* » que les utilisateurs souhaitent utiliser.

2.1. Cas d'étude

Le cas d'étude utilisé dans ce chapitre, est une version simplifiée du processus de gestion d'une clinique « *e-clinique* ».

Dans l'état actuel, une personne représentant un patient peut demander un traitement à la clinique. Afin de commencer le processus, la patient a besoin (i) d'un rendez-vous, (ii) d'un diagnostic de sa maladie, (iii) d'un traitement et (iv) de régler les actes médicaux.

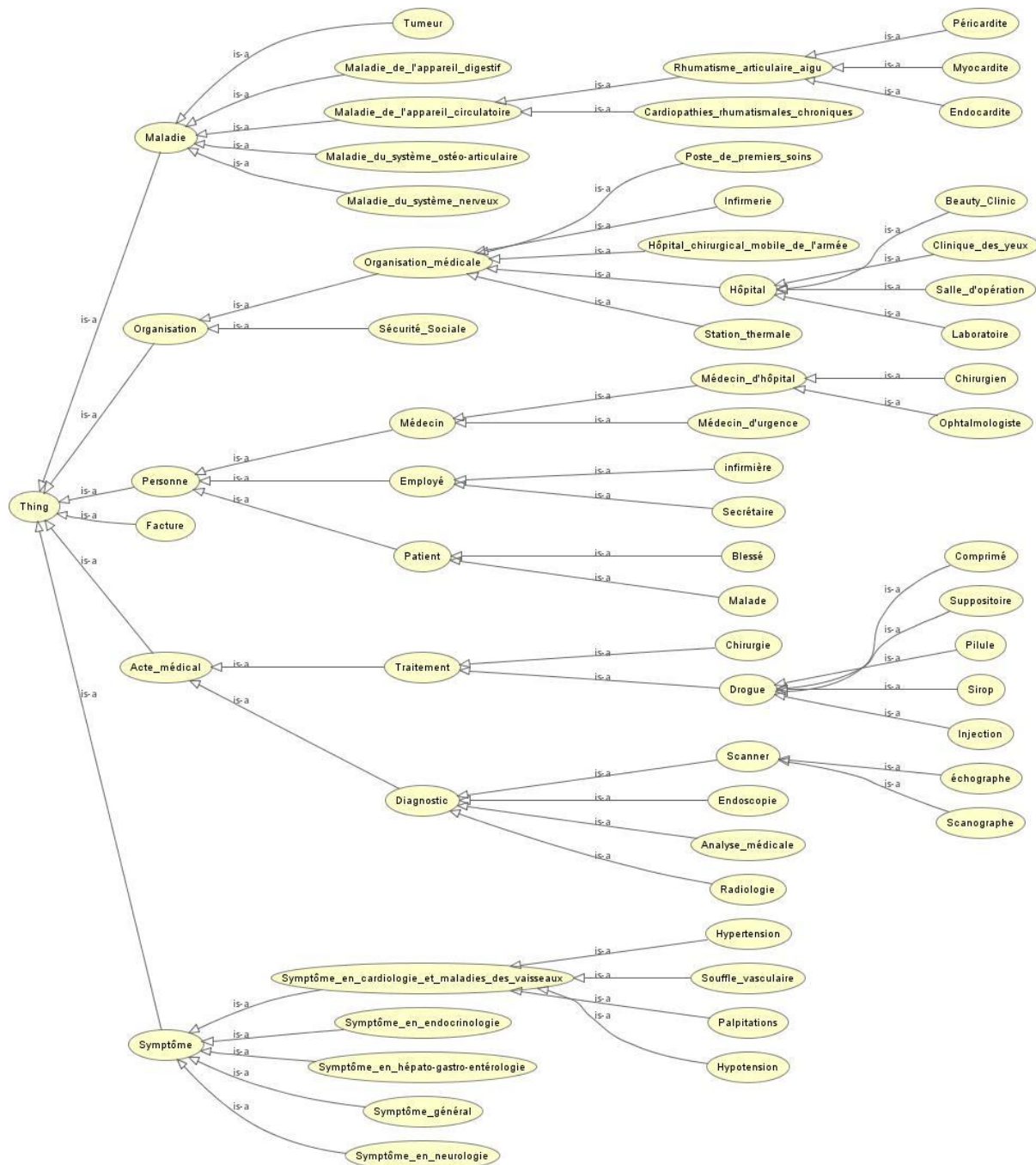


Figure 6.1. L'ontologie de santé

Dans le cas où la demande de rendez-vous est acceptée, le patient effectue les diagnostics nécessaires à la clinique. Après que le patient reçoive les traitements médicaux, les frais sont calculés et présentés à un tiers ou au patient directement pour préparer le paiement.

Puisque nous n'avons pas trouvé d'ontologie complète du domaine de santé, nous en avons construite une (Figure 6.1) en fusionnant six ontologies incomplètes ([Klusch et al., 2006],

Medical Transport Company Ontology, Medical Flight Company Ontology, Hospital Reception Ontology, Hospital physician Ontology et Health Insurance Ontology).

2.2. Analyse des dysfonctionnements

Le processus est long et compliqué sur les patients. On observe qu'un temps considérable est gaspillé à transmettre des documents d'un département à un autre. En outre, il incombe au patient de suivre son dossier.

De plus, de nombreuses activités sont distribuées entre organisations ce qui pose un problème d'hétérogénéité des données.

L'ensemble de ces problèmes a des conséquences sur la performance de la clinique. Le diagnostic de ces dysfonctionnements montre qu'il serait utile d'automatiser le travail administratif de la clinique pour aboutir à un état de fonctionnement satisfaisant. Afin d'améliorer le rendement en termes de patients traités et améliorer la qualité des services fournis aux patients, il serait utile de publier les services de la clinique.

3. PUBLICATION DES SERVICES INTENTIONNELS

Ce paragraphe est consacré à la publication des services intentionnels dans l'annuaire pour ce cas d'application.

Nous illustrons ce cas par étapes. Dans la première étape, le fournisseur utilise le formalisme de la *Carte* [Rolland, 2000] pour la découverte des services intentionnels à partir d'un modèle du business. Ensuite, il spécifie les services intentionnels présentés par les cartes en utilisant le modèle intentionnel de services MIS [Kaabi, 2007]. La deuxième étape consiste à utiliser le processus de publication pour aider le fournisseur dans sa démarche de publication. Finalement, la dernière étape consiste à implémenter les descripteurs de services dans un annuaire étendu.

3.1. La carte « e-clinique »

Cette section a pour but de décrire le processus pour aider les patients à effectuer un examen médical. Ce processus est décrit en termes intentionnels sous forme de carte. La description intentionnelle du processus ainsi produite est représentée par une carte nommée *e-clinique*.

Comme indiqué précédemment, l'objectif du processus s'étend à aider les patients à effectuer un examen médical. Les quatre objectifs principaux assignés au processus sont donc : *accueillir un patient*, *diagnostiquer une maladie*, *prescrire un traitement*, *éditer la facture de l'acte médical* et *régler une facture*. La carte de la Figure 6.2 décrit la manière de procéder pour réaliser les objectifs identifiés. Comme le montre la carte *e-clinique*, un ensemble de buts sont identifiés qui doivent ou peuvent être accomplis afin d'aider les patients à effectuer les actes médicaux ainsi qu'un ensemble de stratégies qui définissent les manières d'accomplir ces buts.

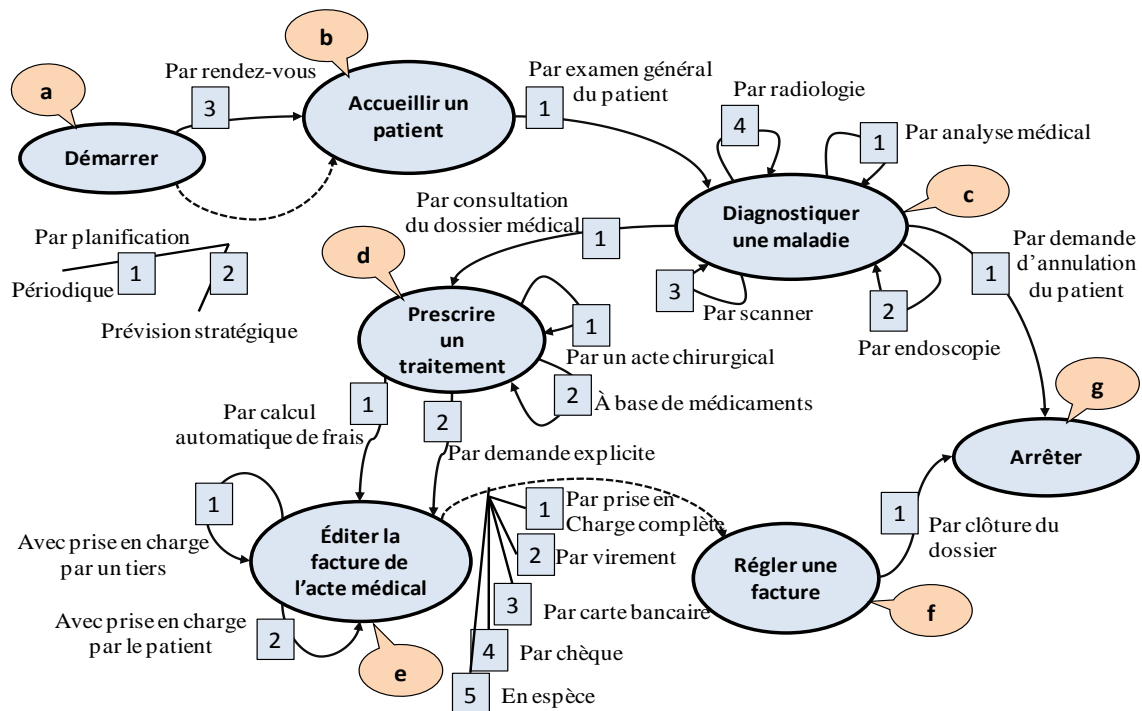


Figure 6.2. La carte e-clinique

3.1.1. Intentions de la carte e-clinique

Les intentions de la carte e-clinique : Démarrer, Accueillir un patient, Enregistrer le dossier médical d'un patient, Elaborer le règlement de l'acte médical, Effectuer un paiement et Arrêter sont détaillées comme suit :

Démarrer : indique le point de départ de l'application du processus.

Accueillir un patient : fait référence à la demande de rendez-vous par le patient. Cette intention vise à analyser les différents besoins des clients, internes et externes, avec tous les moyens possibles et en utilisant plusieurs approches. Ceci fournit les données de base du patient.

Diagnostiquer une maladie : correspond aux examens médicaux qu'un patient subit à la clinique. Les résultats des différents examens médicaux effectués à la clinique sont stockés dans le dossier du patient.

Prescrire un traitement : correspond aux traitements effectués à la clinique. Ces traitements dépendent des résultats des différents examens médicaux obtenus auparavant.

Éditer la facture de l'acte médical : correspond au calcul de frais des examens médicaux et la prise en charge. Ceci fournit les données de base pour le paiement.

Régler une facture : correspond au paiement de frais des examens effectués à la clinique.

Arrêter : le processus peut se terminer par une demande d'annulation ou par clôture de dossier.

3.1.2. Stratégies de la carte e-clinique

Les stratégies possibles pour réaliser l'intention *Accueillir un patient* :

- *Par rendez-vous* : Cette stratégie a été introduite pour permettre au patient d'obtenir un rendez-vous aisément.
- *Par planification (périodique ou prévision stratégique)* : Cette stratégie traite l'accueil des patients soit périodiquement ou bien par prévision stratégique. Cela consiste à planifier l'arrivée des patients afin d'organiser le travail de la clinique.

Pour réaliser l'intention **Diagnostiquer une maladie**, il existe cinq stratégies :

- *Par examen général du patient* : Cette stratégie traite l'ouverture du dossier du patient par les examens physiques de base afin de mise au point de son état général de santé.
- *Par analyse médical* : Cette stratégie traite toutes les analyses médicales que le patient effectue pendant sa visite à la clinique.
- *Par endoscopie* : Cette stratégie permet le diagnostic en utilisant l'endoscopie.
- *Par scanner* : Cette stratégie permet le diagnostic en utilisant le scanner.
- *Par examen radiologique* : Cette stratégie permet d'enregistrer les examens radiologiques du patient et de poser un diagnostic.

Pour réaliser l'intention **Prescrire un traitement**, il existe trois stratégies :

- *Par consultation du dossier médical* : La consultation du dossier médical aide les médecins à décider du traitement selon l'état du patient.
- *Par un acte chirurgical* : Cette stratégie concerne les chirurgies qu'on effectue pour le patient à la clinique.
- *Par la prescription de médicaments* : le patient peut passer directement aux médicaments prescrits par les médecins.

Pour réaliser l'intention **Éditer la facture de l'acte médical**, il existe quatre stratégies :

- *Par calcul automatique de frais* : Cette stratégie traite le calcul des frais des examens médicaux automatiquement.
- *Par demande explicite* : Cette stratégie traite le calcul des frais des examens médicaux suite à la demande du patient.
- *Avec prise en charge par un tiers* : Cette stratégie traite la prise en charge des frais des examens médicaux par un tiers.
- *Avec prise en charge par le patient* : Cette stratégie traite la prise en charge des frais des examens médicaux par le patient.

Pour réaliser l'intention **Régler une facture**, il existe cinq stratégies :

- *Par prise en charge complète* : cette stratégie indique que le paiement a été traité complètement par un tiers.
- *Par virement* : le paiement est effectué par virement bancaire.
- *Par carte bancaire* : le paiement est effectué par carte bancaire.
- *Par chèque* : le paiement est effectué par chèque.
- *En espèce* : le paiement est effectué en espèce.

Pour réaliser l'intention **Arrêter**, il y a deux stratégies :

- *Par demande d'annulation du patient* : Si le patient demande l'annulation du processus. Cela doit faire l'objet d'une demande explicite.
- *Par clôture du dossier* : Le processus se termine normalement par la clôture du dossier et le paiement des frais.

3.2. Identification des services intentionnels dans la carte e-clinique

La génération des services du modèle MiS à partir de la carte *e-clinique* se fait en appliquant le processus, présenté au chapitre 5. Celui-ci est structuré en deux sous étapes :

Etape 1 : Définir les services atomiques.

Etape 2 : Définir les services agrégats.

Le déroulement de chacune de ces sous étapes est détaillé dans les sous sections ci-après de la manière suivante : la sous section 3.2.1 illustre la manière d'identification des services atomiques et la sous section 3.2.2 illustre la manière d'identification des services agrégats.

3.2.1. Etape 1 : Définir les services atomiques

Selon le processus de génération du modèle MiS, les services atomiques sont identifiés en associant à chaque section opérationnalisable de la carte un service atomique.

La carte *e-clinique* compte vingt deux sections opérationnalisables. Par conséquent, nous identifions vingt deux services intentionnels atomiques récapitulés au Tableau 8.1.

Tableau 8.1. Identification des services atomiques à partir de la carte e-clinique

Section de la carte	Service atomique	N°
ab1	<i>S</i> accueillir un patient par planification périodique	<i>S</i> ₁
ab2	<i>S</i> accueillir un patient par planification prévision stratégique	<i>S</i> ₂
ab3	<i>S</i> accueillir un patient par rendez-vous	<i>S</i> ₃
bc1	<i>S</i> diagnostiquer une maladie par examen général	<i>S</i> ₄
cc1	<i>S</i> diagnostiquer une maladie par analyse médicale	<i>S</i> ₅
cc1	<i>S</i> diagnostiquer une maladie par endoscopie	<i>S</i> ₆
cc3	<i>S</i> diagnostiquer une maladie par scanner	<i>S</i> ₇
cc4	<i>S</i> diagnostiquer une maladie par radiologie	<i>S</i> ₈
cd1	<i>S</i> prescrire un traitement par consultation du dossier médical	<i>S</i> ₉
dd1	<i>S</i> prescrire un traitement par un acte chirurgical	<i>S</i> ₁₀
dd2	<i>S</i> prescrire un traitement par la prescription des médicaments	<i>S</i> ₁₁
de1	<i>S</i> éditer la facture de l'acte médical par calcul automatique de frais	<i>S</i> ₁₂
de2	<i>S</i> éditer la facture de l'acte médical par demande explicite	<i>S</i> ₁₃
ee1	<i>S</i> éditer la facture de l'acte médical avec prise en charge par un tiers	<i>S</i> ₁₄
ee2	<i>S</i> éditer la facture de l'acte médical avec prise en charge par le patient	<i>S</i> ₁₅

ef1	S <i>régler la facture par prise en charge complète</i>	S_{16}
ef2	S <i>régler la facture par virement</i>	S_{17}
ef3	S <i>régler la facture par carte bancaire</i>	S_{18}
ef4	S <i>régler la facture par chèque</i>	S_{19}
ef5	S <i>régler la facture en espèce</i>	S_{20}
fg1	S <i>arrêter par clôture du dossier</i>	S_{21}
cgl	S <i>arrêter par demande d'annulation du client</i>	S_{22}

3.2.2. Etape 2 : Définir les services agrégats

Afin de définir les services agrégats, le processus de génération du modèle MiS définit des services agrégats en appliquant deux sous étapes :

- (i) Génération de tous les chemins de la carte et
- (ii) Identification des services agrégats.

Les deux sous sections suivantes décrivent l'application de chacune de ces sous étapes.

3.2.2.1. Génération de tous les chemins de la carte

Les enchaînements des sections dans une carte identifient diverses sortes de services. Cependant, il est difficile d'identifier toutes les possibilités de manière exhaustive. MiS utilise l'algorithme de MacNaughton et Yamada [MacNaughton et Yamada, 1960] et l'applique au modèle de la carte afin d'identifier tous les chemins possibles entre un but source *Démarrer* et un but cible *Arrêter*.

Durant cette étape, les chemins de la carte *e-clinique* sont décrits en premier lieu par des formules mathématiques propres au formalisme de l'algorithme. Par la suite, des mises en correspondance sont appliquées afin de définir les relations entre les sections de la carte.

La formule décrivant tous les chemins entre le but a et le but g de la carte *e-clinique* est :

$$Y_{a,\{a,b,c,d,e,f,g\},g} = \bullet (X_{ab}, X_{bc}, X_{cc}^*, U(\bullet(X_{cd}, X_{dd}^*, X_{de}, X_{ee}^*, X_{ef}, X_{fg}), X_{cg}))$$

Cette formule nous indique que pour aller de a à g , il y a une séquence à suivre. En effet, la formule X_{ab} est utilisée pour satisfaire le but b à partir du but a . La formule X_{bc} est utilisée pour satisfaire le but c à partir du but b . Ensuite, X_{cc} est utilisée pour aller de c vers c . Pour aller de c à g , il faut choisir l'un des deux chemins suivants : soit aller de c à g directement (cette possibilité est décrite par la formule X_{cg}), soit aller de c à d puis de d à d puis de d à e puis de e à e puis de e à f et enfin de f à g (ce second chemin est exprimé par la formule $\bullet(X_{cd}, X_{dd}^*, X_{de}, X_{ee}^*, X_{ef}, X_{fg})$).

Spécialisation des X_{st} en sections de la carte *e-clinique* et relations entre elles

A ce niveau, nous spécialisons les formules X_{st} obtenues en termes de sections de la carte et les relations entre elles à savoir : chemin, multi-chemin, paquet et multi-segment. Ceci est mis en place en appliquant les règles suivantes :

- ✓ Affecter à chaque formule X_{st} , la section, le paquet ou le multi-segment correspondant.
- ✓ Une formule X_{st} est un chemin de la carte si elle ne correspond pas à une section, un paquet ou un multi-segment. Ce chemin est constitué d'un ensemble de

formules reliées par un lien de séquence si l'ordre d'exécution est prédéfini ou bien un lien parallèle dans le cas contraire.

- ✓ Un ensemble de formules reliées par le symbole de composition « • » définit un chemin constitué de sections, paquets ou multi-segments reliés par un lien de séquence.
- ✓ Un ensemble de formules reliées par le symbole union « \cup » définit un multi-chemin. Chaque formule, ou ensemble de formules, précédées du symbole « \cup » peut être composée de sections, paquets, multi-segments et chemins.
- ✓ Une formule X_{st} itérative (ayant le symbole « * ») identifie par défaut une section, paquet, multi-segment, chemin ou multi-chemin itératif et optionnel sauf s'il y a décision pour les rendre obligatoire ou itératif.

Par application des règles énoncées précédemment sur la carte *e-clinique* (cf. Figure 6.1), nous obtenons les formules illustrées dans le Tableau 8.2 suivant :

Tableau 8.2. Liste des relations entre les sections de la carte *e-clinique*

Type de la relation	Relation identifiée
Paquet	$P_{de} = \otimes (de_1, de_2)$ $P_{ef} = \otimes (ef_1, ef_2, ef_3, ef_4, ef_5)$
Multi-segments	$MS_{ab} = \vee (ab_1, ab_2, ab_3)$ $MS_{cc} = \vee (cc_1, cc_2, cc_3, cc_4)$ $MS_{dd} = \vee (dd_1, dd_2)$ $MS_{ee} = \vee (ee_1, ee_2)$
Chemin	$C_{e,\{f\},g} = \bullet (MS_{ee}^*, P_{ef}, fg_1)$ $C_{d,\{e,f\},g} = \bullet (P_{de}, C_{e,\{f\},g})$ $C_{c,\{d,e,f\},g} = \bullet (cd_1, MS_{dd}^*, C_{d,\{e,f\},g})$ $C_{a,\{b,c,d,e,f\},g} = \bullet (MS_{ab}, bc_1, MS_{cc}^*, MC_{c,\{d,e,f\},g})$
Multi-chemin	$MC_{c,\{d,e,f\},g} = \cup (cg_1, C_{c,\{d,e,f\},g})$

Nous pouvons vérifier les résultats obtenus dans le Tableau 8.2 à partir de la carte *e-clinique*. Le chemin $C_{a,\{b,c,d,e,f\},g}$ définit toutes les compositions possibles de services entre le but source *Démarrer* (ayant le code *a*) et le but cible *Arrêter* (ayant le code *g*). Au début, il y a un choix d'une ou plusieurs possibilités du multi-segment MS_{ab} pour atteindre le but *b*. L'étape suivante consiste à choisir la seule possibilité bc_1 pour atteindre le but *c*. L'étape suivante consiste à choisir une ou plusieurs possibilités du multi-segment MS_{cc} pour atteindre le but *c*. Ensuite, il y a deux chemins pour satisfaire le but *g* à partir du but *c* qui sont décrits par le multi-chemin $MC_{c,\{d,e,f\},g}$. Le premier chemin consiste à choisir cg_1 . Le deuxième chemin est $C_{c,\{d,e,f\},g}$ qui suggère d'atteindre le but *d* en choisissant la seule possibilité cd_1 . L'étape suivante consiste à choisir une ou plusieurs possibilités du multi-segment MS_{dd} , pour atteindre le but *e*. une alternative du paquet P_{de} , une alternative du paquet P_{dd} puis atteindre le but *f* par le chemin ef_1 , une alternative du paquet P_{ef} , puis réaliser fg_1 .

3.2.2.2. Identification des services agrégats

Nous proposons dans cette section, d'identifier les services agrégats à l'aide d'un ensemble de règles présentées en détail dans [Kaabi, 2007]. Ces règles permettent la mise en correspondance entre les relations entre les sections de la carte *e-clinique* et les services correspondants.

Les règles à appliquer consistent à (i) *affecter à chaque chemin de la carte un service composite séquentiel*, (ii) *affecter à chaque multi-chemin de la carte un service à variation de chemin*, (iii) *affecter à chaque paquet de la carte un service à choix alternatif*, et (iv) *affecter à chaque multi-segment de la carte un service à choix multiple*.

L'application de cette étape sur la carte *e-clinique* donne les services agrégats résumés au Tableau 8.3.

Tableau 8.3. Liste des services agrégats générés à partir de la carte *e-clinique*

Type de relation		Service identifié		N°
Paquet	$P_{de} = \otimes (de_1, de_2)$	Service à choix alternatif	$S_{Préparer la facture de l'acte médical}$	S_{23}
	$P_{ef} = \otimes (ef_1, ef_2, ef_3, ef_4, ef_5)$		$S_{Régler une facture}$	S_{24}
Multi-segments	$MS_{ab} = \vee (ab_1, ab_2, ab_3)$	Service à choix multiple	$S_{Accueillir un patient}$	S_{25}
	$MS_{cc} = \vee (cc_1, cc_2, cc_3, cc_4)$		$S_{Diagnostiquer une maladie d'un patient}$	S_{26}
	$MS_{dd} = \vee (dd_1, dd_2)$		$S_{Prescrire un traitement d'un patient}$	S_{27}
	$MS_{ee} = \vee (ee_1, ee_2)$		$S_{Éditer la facture de l'acte médical}$	S_{28}
Chemin	$C_{e,\{f\},g} = \bullet (MS_{ee}^*, P_{ef}, fg_1)$	Service composite séquentiel	$S_{Régler la facture de l'acte médical}$	S_{29}
	$C_{d,\{e,f\},g} = \bullet (P_{de}, C_{e,\{f\},g})$		$S_{Facturer un acte médical}$	S_{30}
	$C_{c,\{d,e,f\},g} = \bullet (cd_1, MS_{dd}^*, C_{d,\{e,f\},g})$		$S_{Réaliser une visite médicale d'un patient}$	S_{31}
	$C_{a,\{b,c,d,e,f\},g} = \bullet (MS_{ab}, bc_1, MS_{cc}^*, MC_{c,\{d,e,f\},g})$		$S_{Traiter un patient dans la clinique}$	S_{32}
Multi-chemin	$MC_{c,\{d,e,f\},g} = \cup (cg_1, C_{c,\{d,e,f\},g})$	Service à variation de chemin	$S_{Finaliser une visite médicale d'un patient}$	S_{33}

Le service agrégat qui permet de traiter un patient à la clinique est $S_{traiter un patient à la clinique}$. Ce service est une composition de quatre services : $S_{accueillir un patient}$, $S_{diagnostiquer une maladie par examen général}$, $S_{diagnostiquer une maladie d'un patient}$ et $S_{finaliser une visite médicale d'un patient}$. Le premier service est un service à choix multiple qui permet de formuler la demande de rendez-vous du patient à travers la réalisation du but *Accueillir un patient*. Ceci consiste à

- Accueillir le patient par rendez-vous normal $S_{Accueillir le patient par rendez-vous}$ ou bien
- Accueillir le patient par planification périodique ou par provision stratégique $S_{Accueillir le patient par planification périodique}$ ou $S_{Accueillir le patient par provision stratégique}$

Le deuxième service $S_{\text{Diagnostiquer une maladie par examen général}}$ est un service atomique qui permet à la clinique de commencer le diagnostic du patient par un examen physique de l'état général à travers la réalisation du but *Diagnostiquer une maladie*.

Le troisième service $S_{\text{Diagnostiquer une maladie d'un patient}}$ est un service à variation multiple qui permet à la clinique de réaliser les diagnostics pour un patient à travers la réalisation du but *Diagnostiquer une maladie*. Ceci consiste à :

- Diagnostiquer une maladie par effectuer toutes les analyses médicales pour le patient à la clinique $S_{\text{Diagnostiquer une maladie par analyse médical}}$,
- Diagnostiquer une maladie par effectuer une endoscopie pour le patient à la clinique $S_{\text{Diagnostiquer une maladie par endoscopie}}$,
- Diagnostiquer une maladie par effectuer un scanner pour le patient à la clinique $S_{\text{Diagnostiquer une maladie par scanner}}$,
- Diagnostiquer une maladie par effectuer toutes les radiologies pour le patient à la clinique $S_{\text{Diagnostiquer une maladie par radiologie}}$,

Enfin, le quatrième service $S_{\text{Finaliser le suivi médical d'un patient}}$ est un multi-chemin. Le premier chemin consiste à exécuter le service atomique $S_{\text{Arrêter par demande d'annulation du client}}$. Ceci consiste à arrêter le processus par le patient en annulant la demande. Le deuxième chemin consiste à réaliser une visite du patient à l'aide du service composite séquentiel $S_{\text{Réaliser une visite médicale d'un patient}}$. Ce service est une composition de trois services : $S_{\text{Prescrire un traitement par consultation du dossier médical}}$, $S_{\text{Prescrire un traitement d'un patient}}$ et $S_{\text{Facturer un acte médical}}$.

$S_{\text{Prescrire un traitement par consultation du dossier médical}}$ est un service atomique et le service $S_{\text{Prescrire un traitement d'un patient}}$ est un service à choix multiple qui permet de réaliser le but *Prescrire un traitement*. Ceci consiste à choisir l'une des services suivants :

- Traiter le patient par la prescription d'un acte chirurgical à la clinique $S_{\text{Prescrire un traitement Par un acte chirurgical}}$,
- Traiter le patient par la prescription des médicaments effectuée pendant la visite du patient à la clinique $S_{\text{Prescrire un traitement Par la prescription de médicaments}}$,

Le service $S_{\text{Facturer un acte médical}}$ est un service composite séquentiel qui consiste à réaliser la facturation de l'acte médical, ce service est une composition de deux services : $S_{\text{Préparer la facture de l'acte médical}}$ et $S_{\text{Régler la facture de l'acte médical}}$.

Le service $S_{\text{Editer la facture de l'acte médical}}$ est un service à choix alternatif qui consiste à choisir l'une des alternatives suivantes :

- Préparer la facture par le calcul automatiquement $S_{\text{Editer la facture de l'acte médical par calcul automatique des frais}}$ ou
- Préparer la facture par la demande du patient $S_{\text{Editer la facture de l'acte médical par demande explicite}}$

Le service $S_{\text{Régler la facture de l'acte médical}}$ est un service composite séquentiel qui consiste à réaliser le règlement de l'acte médical, ce service est une composition de trois services : $S_{\text{Editer la facture de l'acte médical}}$, $S_{\text{Régler une facture}}$ et le service atomique $S_{\text{Arrêter par clôture du dossier}}$.

Le service $S_{\text{Editer la facture de l'acte médical}}$ est un service à choix multiple qui permet de réaliser le but *Editer la facture de l'acte médical*. Ceci consiste à choisir l'une des services suivants :

- Editer la facture avec prise en charge par un tiers $S_{\text{Editer la facture de l'acte médical avec prise en charge par un tiers}}$, ou
- Editer la facture avec prise en charge par le patient $S_{\text{Editer la facture de l'acte médical avec prise en charge par le patient}}$.

Le service S *Régler une facture* est un service à choix alternatif qui consiste à choisir l'une des alternatives suivantes :

- Régler la facture par prise en charge complète S *Régler une facture par prise en charge complète* ou
- Régler la facture par virement S *Régler une facture par virement* ou
- Régler la facture par carte bancaire S *Régler une facture par carte bancaire* ou
- Régler la facture par chèque S *Régler une facture par chèque* ou
- Régler la facture en espèce S *Régler une facture en espèce*

3.3. Déroulement du processus de publication

Nous avons exécuté la carte Map-Publication dans le cas d'*e-clinique*. Toutes les sections de la carte Map-Publication ont été utilisées pour former des chemins de *Démarrer* à *Arrêter*. Pour Annoter le service, on a donc utilisé les directives DRI_1 , DSS_1 et DRI_7

Toutes les autres sections de la carte liées à la directive DSS_1 ont été utilisées au moins une fois (par application des DRI_2 , DRI_3 , DRI_4 , DRI_5 et DRI_6). La Figure 6.3 met en évidence les sections utilisées dans la publication.

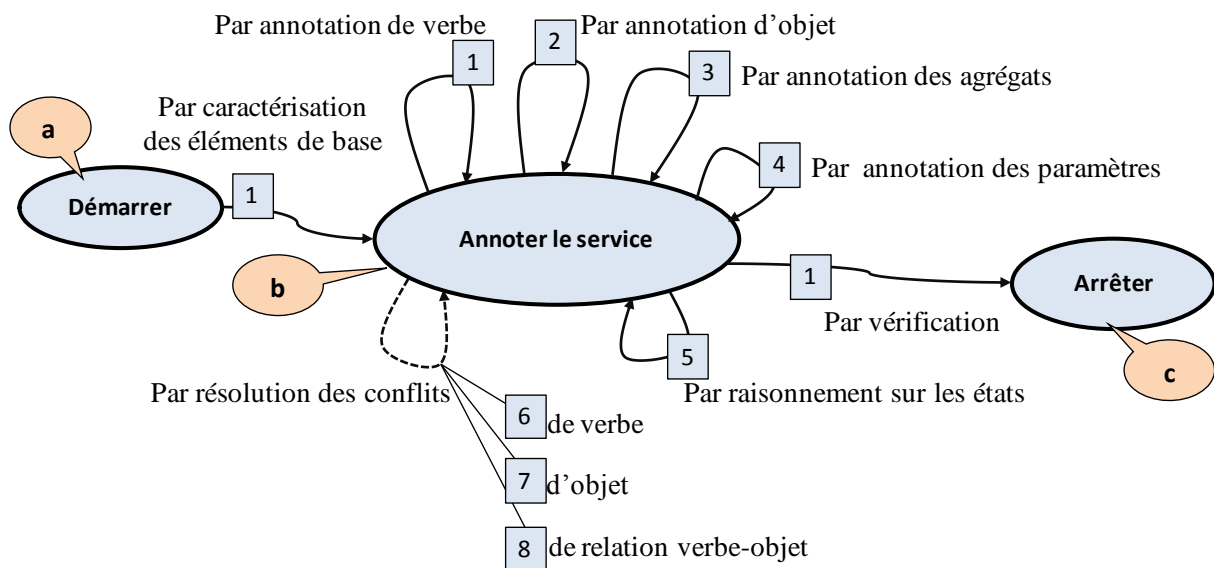


Figure 6.3. Rappel de la carte Map-Publication

3.3.1. Caractérisation des éléments de base des services intentionnels

Après avoir identifié les services intentionnels et leur classification en atomiques ou agrégats, Nous procédons maintenant à appliquer la directive DRI_1 sur les services intentionnels identifiés dans la section précédente. Eventuellement, nous allons déterminer le but associé à chaque service intentionnel et les autres concepts de services liés à l'ontologie de services intentionnels.

3.3.1.1. Formulation des buts de services intentionnels

Tel que présenté dans le chapitre 3, le but est composé d'un verbe, d'une cible, et des paramètres. Par exemple, le but du service intentionnel « *accueillir un patient par rendez-vous* » est formalisé comme :

(Accueillir patient par rendez-vous) But (S₁) Service atomique

Le tableau 8.4 présente la formulation des buts de services de la carte *e-clinique*.

Tableau 8.4. *La formulation des buts de services de la carte e-clinique*

Service	But
S ₁	(accueillir un patient par planification périodique) <i>But (S₁) Service atomique</i>
S ₂	(accueillir un patient par prévision stratégique) <i>But (S₂) Service atomique</i>
S ₃	(accueillir un patient par rendez-vous) <i>But (S₃) Service atomique</i>
S ₄	(diagnostiquer une maladie par examen général) <i>But (S₄) Service atomique</i>
S ₅	(diagnostiquer une maladie par analyse médicale) <i>But (S₅) Service atomique</i>
S ₆	(diagnostiquer une maladie par endoscopie) <i>But (S₆) Service atomique</i>
S ₇	(diagnostiquer une maladie par scanner) <i>But (S₇) Service atomique</i>
S ₈	(diagnostiquer une maladie par radiologique) <i>But (S₈) Service atomique</i>
S ₉	(prescrire un traitement par consultation du dossier médical) <i>But (S₉) Service atomique</i>
S ₁₀	(prescrire un traitement par un acte chirurgical) <i>But (S₁₀) Service atomique</i>
S ₁₁	(prescrire un traitement par la prescription des médicaments) <i>But (S₁₁) Service atomique</i>
S ₁₂	(éditer la facture de l'acte médical par calcul automatique de frais) <i>But (S₁₂) Service atomique</i>
S ₁₃	(éditer la facture de l'acte médical par demande explicite) <i>But (S₁₃) Service atomique</i>
S ₁₄	(éditer la facture de l'acte médical avec prise en charge par un tiers) <i>But (S₁₄) Service atomique</i>
S ₁₅	(éditer la facture de l'acte médical avec prise en charge par le patient) <i>But (S₁₅) Service atomique</i>
S ₁₆	(régler la facture par prise en charge complète) <i>But (S₁₆) Service atomique</i>
S ₁₇	(régler la facture par virement) <i>But (S₁₇) Service atomique</i>
S ₁₈	(régler la facture par carte bancaire) <i>But (S₁₈) Service atomique</i>
S ₁₉	(régler la facture par chèque) <i>But (S₁₉) Service atomique</i>
S ₂₀	(régler la facture en espèce) <i>But (S₂₀) Service atomique</i>
S ₂₁	(arrêter la visite par clôture du dossier) <i>But (S₂₁) Service atomique</i>
S ₂₂	(arrêter la visite par demande d'annulation du client) <i>But (S₂₂) Service atomique</i>
S ₂₃	(préparer la facture de l'acte médical) <i>But (S₂₃) Service agrégat</i>
S ₂₄	(régler une facture) <i>But (S₂₄) Service agrégat</i>
S ₂₅	(accueillir un patient) <i>But (S₂₅) Service agrégat</i>
S ₂₆	(diagnostiquer une maladie d'un patient) <i>But (S₂₆) Service agrégat</i>
S ₂₇	(prescrire un traitement d'un patient) <i>But (S₂₇) Service agrégat</i>
S ₂₈	(éditer la facture de l'acte médical) <i>But (S₂₈) Service agrégat</i>
S ₂₉	(régler la facture de l'acte médical) <i>But (S₂₉) Service agrégat</i>
S ₃₀	(facturer un acte médical) <i>But (S₃₀) Service agrégat</i>
S ₃₁	(réaliser une visite médicale d'un patient) <i>But (S₃₁) Service agrégat</i>
S ₃₂	(traiter un patient dans la clinique) <i>But (S₃₂) Service agrégat</i>

S ₃₃	(finaliser une visite médicale d'un patient) But (S ₃₃) Service agrégat
-----------------	---

3.3.2. Annotation de services intentionnels

Dans cette section, nous allons appliquer la directive DSS₁ sur les services des tableaux 8.1 et 8.3. Ceci consiste d'abord, à coupler les éléments du but de services avec les éléments correspondant de l'ontologie *iSOnto*. Le couplage est réalisé en identifiant la nature de chaque élément de but et en le reliant avec l'élément correspondant.

3.3.2.1. Annotation de verbe

On utilise la directive DRI₂ d'abord, pour s'assurer du sens de verbe, parce qu'un verbe normalement appartient à une ou plusieurs classes de sens de verbe. Par conséquent, cette directive aide l'utilisateur, dans la précision de la classe de sens de verbe. Par exemple, le verbe « diagnostiquer » appartient à la classe de sens de verbe « déterminer ou de distinguer la nature d'un problème ou une maladie à travers une analyse diagnostique » qui contient le verbe « analyser »... Le verbe « prescrire » appartient à la classe de sens de verbe de « exécuter des commandes ou des commandes pour » qui contient les verbes « dicter », « ordonner »... Le verbe « accueillir » appartient à la classe de sens de verbe d'accueil qui contient les verbes « recevoir », « accueillir »... (Figure 6.4). Le verbe « régler » appartient à la classe de sens de verbe « donner de l'argent, généralement en échange de biens ou services » qui contient les verbes « payer », « verser », « acquitter », « rembourser »...

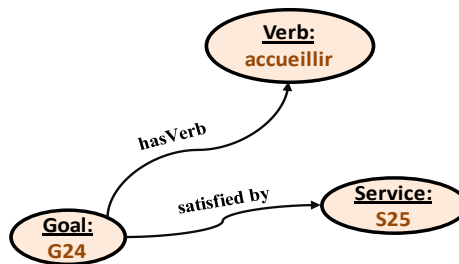


Figure 6.4. Annotation de verbe

Le résultat de l'application de cette directive est le « verbe » que nous ajoutons à l'annotation du descripteur.

3.3.2.2. Annotation d'objet

DRI₃ consiste à identifier l'objet dans le but et à vérifier son appartenance à l'ontologie des produits. Dans cette directive, on prend le concept d'objet, le localise dans l'ontologie des produits, puis on rend tous les concepts adjacents à cet objet pour donner au fournisseur la possibilité de remplacer l'objet avec d'autres concepts similaires. Par exemple, dans le service S₂₅, le produit « patient » est une instanciation de « personne », une généralisation de « malade » et « blessé » ou proche de « docteur », « infirmière », « employé »... (Figure 6.5).

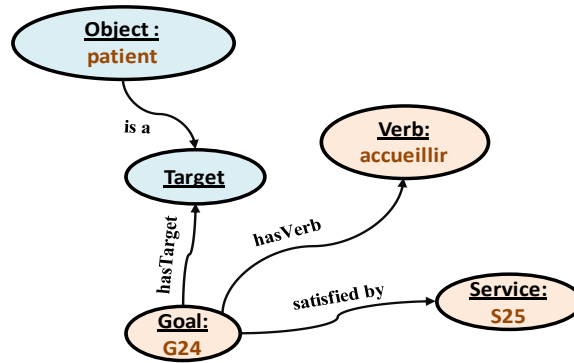


Figure 6.5. Annotation de l'objet

L'application de cette directive a permis de choisir « l'objet » que nous ajoutons à l'annotation du descripteur.

3.3.2.3. Annotation des agrégats

L'annotation des agrégats consiste d'abord à ajouter les composants du service (*type du service agrégat, components*). La complétude est réalisée conformément à l'ontologie de services intentionnels *iSOnto*. En appliquant DRI_4 pour le service agrégat S_{25} , on ajoute que le type du service est « *aggregate* » ou plus précisément variante de type « *Multithread* », (Figure 6.6).

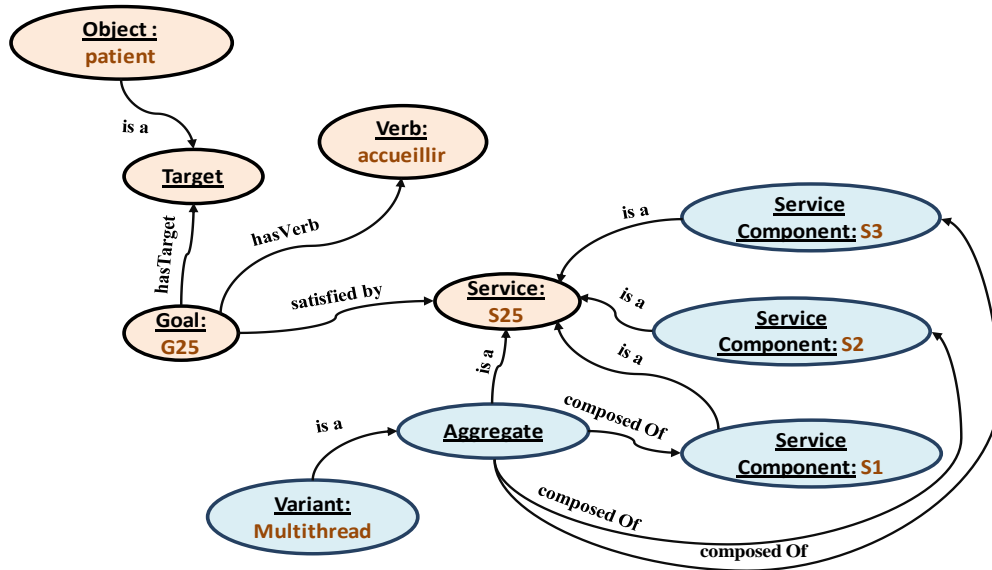


Figure 6.6. Annotation des agrégats

Dans ce cas, nous ajoutons les composants de ce service, qui sont les services atomiques S_1 , S_2 , et S_3 à l'annotation du descripteur.

3.3.2.4. Raisonnement sur les états

Le raisonnement sur l'état dépend du verbe comme le montre la directive DRI_6 :

- « Accueillir » est un verbe abstrait intellectuel de service, il accepte tout les paramètres (temps, location). Il accepte aussi le changement de situations état final = 'Patient' (Figure 6.7).

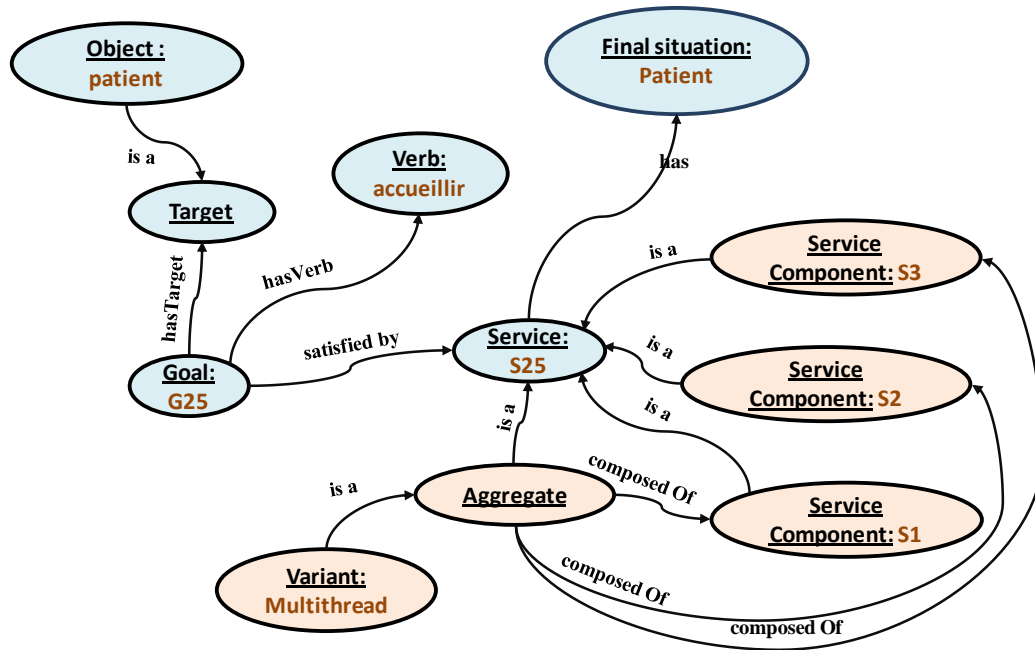


Figure 6.7. Raisonnement sur les états

En appliquant cette directive, nous ajoutons la « *situation finale* » à l'annotation du descripteur.

3.3.2.5. Résolution des conflits

Trois types de conflits sont identifiés dans le Map-Publication. Mais, ils concernent la relation entre le verbe et l'objet.

- ✓ (diagnostiquer)-(une maladie)
- ✓ (Accueillir)-(patient)
- ✓ (Régler)-(facture)

En vérifiant dans l'ontologie des produits *pOnto*, nous avons trouvé que chaque objet a un lien avec son verbe. Nous acceptons dans ce cas que les couples verbe-objet sont tous compatibles.

3.3.2.6. Représentation des services intentionnels dans iSOnto

Avant de passer à l'annotation des services, nous présentons les concepts de services liés à l'ontologie de services intentionnels. La Figure 6.8, introduit la présentation du service atomique *S Diagnostiquer une maladie par examen général (S4)* dans *iSOnto*. Comme on peut le voir sur cette figure, nous avons fait une distinction entre les nœuds et les feuilles lors de l'instanciation de l'ontologie *iSOnto* en couleur brune.

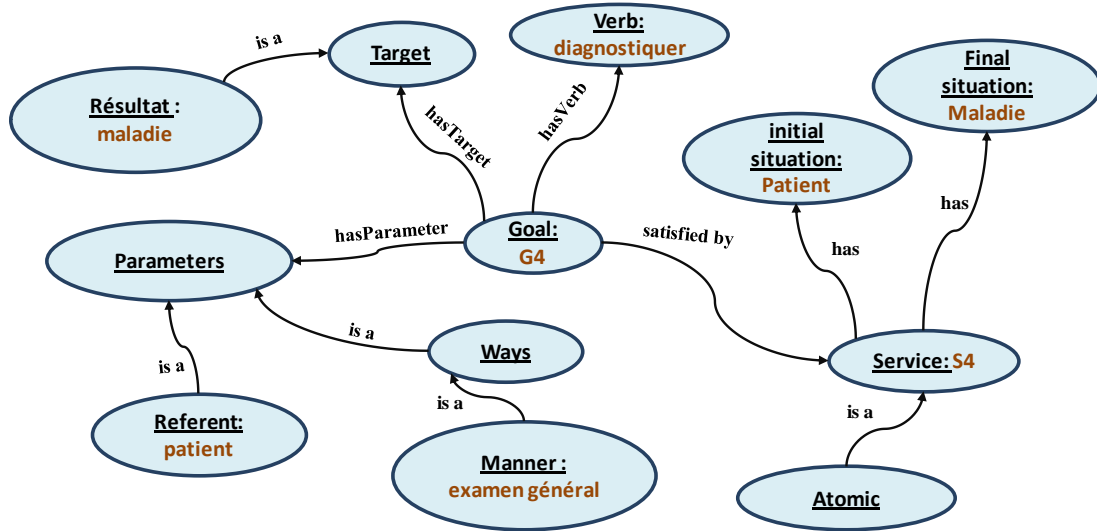


Figure 6.8. Présentation du service atomique $S_{\text{Diagnostiquer une maladie par examen général}}$ dans iSOnTo

Dans la Figure 6.9, nous introduisons la présentation du service à variation $S_{\text{accueillir un patient}}$ (S_{25}). Ce service agrégat spécifie deux alternatives pour accueillir un patient : soit par « *par rendez-vous* » ou par « *planification* ».

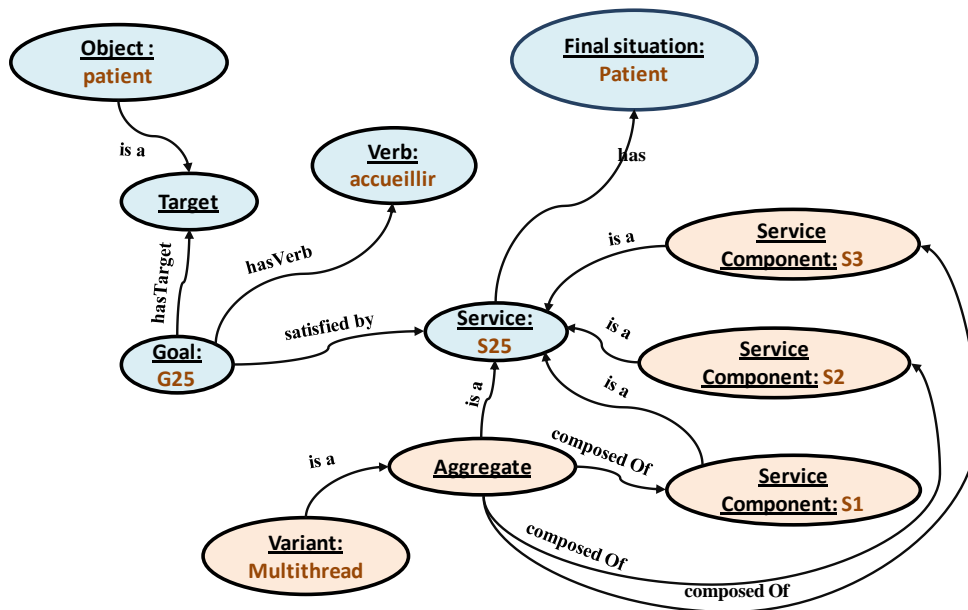


Figure 6.9. Présentation du service à variation $S_{\text{accueillir un patient}}$ dans iSOnTo

Et à la Figure 6.10, nous introduisons la présentation du service composite $S_{\text{Régler la facture}}$ (S_{29}). Ce service intentionnel agrégat est composé d'une séquence de trois services intentionnels : d'un service à choix multiple $S_{\text{Editer la facture de l'acte médical}}$, d'un service à choix alternatif $S_{\text{Régler une facture}}$ et un service intentionnel atomique $S_{\text{arrêter par clôture du dossier}}$.

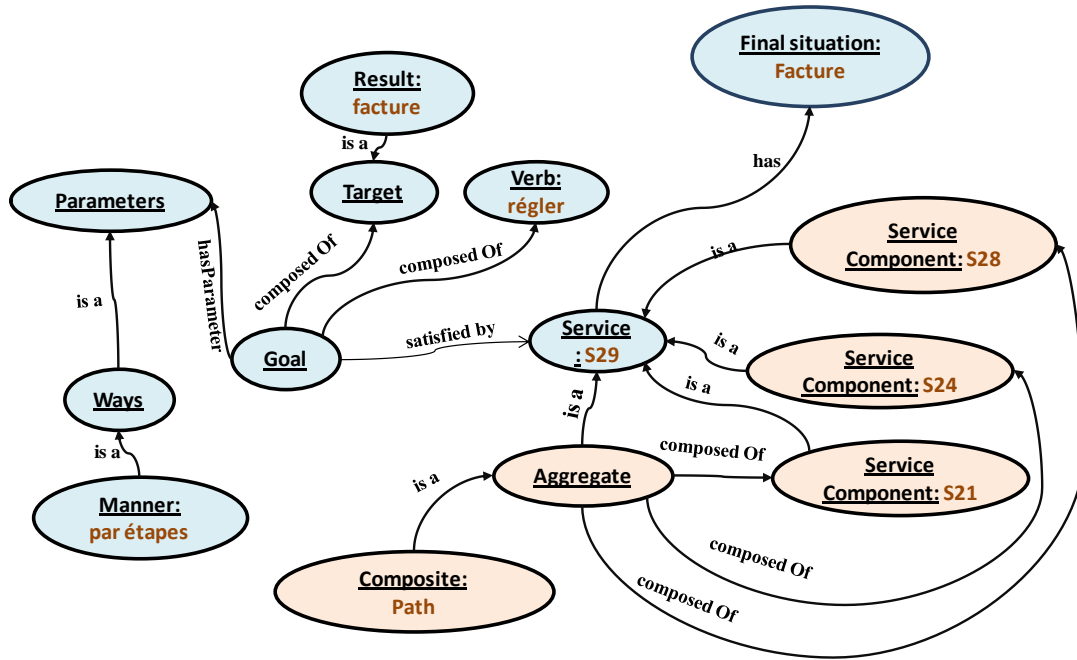


Figure 6.10. Présentation du service composite $S_{\text{Régler la facture}}$ dans $i\text{Onto}$

Les trois dernières figures représentent les couples (nom, valeur) à ajouter dans le descripteur de chaque service. Chaque couple représente un concept d' $i\text{Onto}$ et sa valeur dans les deux ontologies $v\text{Onto}$ et $p\text{Onto}$. Cette représentation graphique est traduite dans la suite en annotations dans les descripteurs de base.

Le tableau 8.5 contient les trois services précédents : atomique $S_{\text{Diagnostiquer une maladie par examen général}}$, composite $S_{\text{Régler la facture}}$ et à variation $S_{\text{accueillir un patient}}$ en utilisant les concepts d' $i\text{Onto}$, de la même manière qu'ils vont être présentés dans le descripteur.

Tableau 8.5. Les éléments de base des services intentionnels

S_4 : (diagnostiquer) verbe (une maladie) résultat (d'un patient) référent (par examen général) manière (S4) service atomique
S_{25} : (accueillir) verbe (un patient) produit (S25) variation (S1) service composant (S2) service composant (S3) service composant
S_{29} : (Régler) verbe (la facture) résultat (S29) composite (S28) service composant (S24) service composant (S21) service composant

3.3.3. Vérification des annotations

La vérification des annotations se fait d'un côté, par s'assurer que les éléments utilisés respectent les concepts des trois ontologies : $i\text{Onto}$, $v\text{Onto}$ et $p\text{Onto}$. De l'autre côté, par la complétude de l'annotation. Dans notre exemple, nous avons vérifié que les annotations

n'utilisent que les concepts des trois ontologies précédentes et que les annotations sont complètes.

3.4. Descripteurs des services intentionnels en SAWSDL

Nous présentons uniquement la partie modifiée du descripteur en utilisant SAWSDL afin de présenter l'annotation ajoutée aux services dans le tableau 8.5. Par conséquent, le descripteur du service atomique $S_{\text{Diagnostiquer une maladie par examen général}}$ est présenté dans le tableau 8.6.

Tableau 8.6. Le descripteur du service atomique $S_{\text{enregistrer le dossier médical d'un patient par examens}}$

```
<service name="Diagnostiquer une maladie par examen général"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#Diagnostiquer
    &http://.../HealthOntology;#maladie
    &http://.../HealthOntology;#patient
    &http://.../HealthOntology;#examen_medical
    &http://.../HealthOntology;#patient
    &http://.../HealthOntology;#S4">
</service name>
```

Le descripteur du service variante à variation $S_{\text{Accueillir Patient}}$ est présenté dans le tableau 8.7.

Tableau 8.7. Le descripteur du service à variation $S_{\text{Accueillir Patient}}$

```
<service name="AccueillirPatient"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#Product
    &http://.../iServiceOntology;#Multithread
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#final_Sta
  modelReference="&http://.../VerbOntology;#accueillir
    &http://.../HealthOntology;#Patient
    &http://.../HealthOntology;#S25
    &http://.../HealthOntology;#S1
    &http://.../HealthOntology;#S2
    &http://.../HealthOntology;#S3
    &http://.../HealthOntology;#patient">
</service name>
```

Enfin, le descripteur du service composite $S_{\text{Régler la facture}}$ est présenté dans le tableau 8.8.

Tableau 8.8. Le descripteur du service composite $S_{\text{Régler la facture}}$

```
<service name="Régler la facture"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#Product
    &http://.../iServiceOntology;#Path
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#final_stat
    &http://.../iServiceOntology;#manière
  modelReference="&http://.../VerbOntology;#régler
    &http://.../HealthOntology;#facture
    &http://.../HealthOntology;#S29
    &http://.../HealthOntology;#S28
    &http://.../HealthOntology;#S24
    &http://.../HealthOntology;#S21
    &http://.../HealthOntology;#facture
    &http://.../HealthOntology;#par étapes">
</service name>
```

4. RECHERCHE DES SERVICES INTENTIONNELS

Ce paragraphe présente l'étude de cas d'un patient qui cherche à être reçu dans une clinique en utilisant les services intentionnels de l'application *e-clinique*.

Deux scénarios sont introduits : le premier concerne l'accueil d'une personne à la clinique, cette requête est formulée par « *Recevoir une personne à la clinique* ». Le deuxième scénario est le suivant : une personne demande à passer un examen médical à la clinique. Afin de commencer le processus, la personne cherche un service qui peut l'aider à effectuer cet examen avec la possibilité que l'assureur verse directement les frais.

Dans ce cas la personne peut formuler sa requête de la façon suivante :

Passer un examen médical à la clinique en payant les frais par un assureur.

4.1. Déroulement du processus de recherche

Nous avons exécuté la carte Map-Recherche (Figure 6.11) en supposant que les services d'*e-clinique* sont disponibles dans l'annuaire. Toutes les sections de la carte Map-Recherche ont été utilisées pour former des chemins de *Démarrer* à *Arrêter*. Pour *formuler une requête* et *proposer des services* en passant par l'affinement, on a donc utilisé toutes les directives.

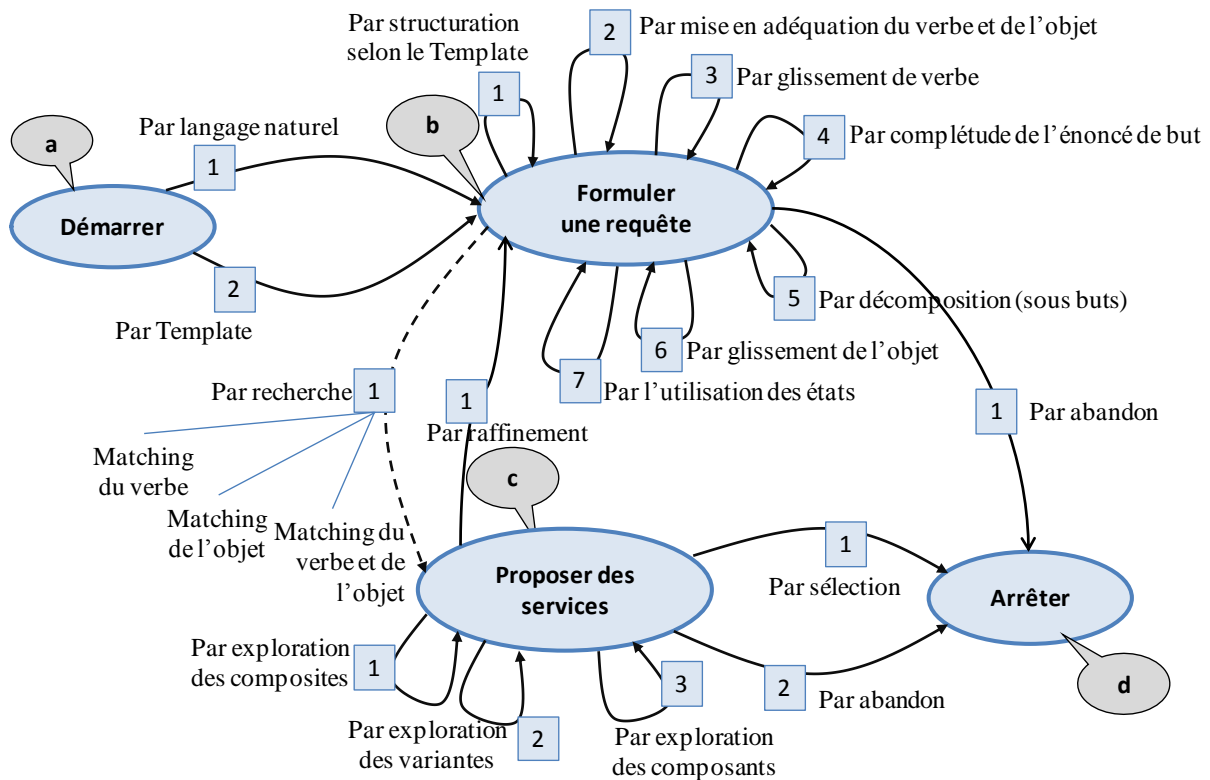


Figure 6.11. Rappel de la carte Map-Recherche

4.1.1. Premier scénario

4.1.1.1. Formulation de requête par template

Tel que présenté au chapitre 3, le but est composé d'un verbe, d'un objet, et de paramètres. Par exemple, la requête « *Recevoir une personne à la clinique* » est formalisée en utilisant la directive DRI_2 , comme :

(Recevoir) *verbe* *(une personne)* *objet* *à (la clinique)* *Location*.

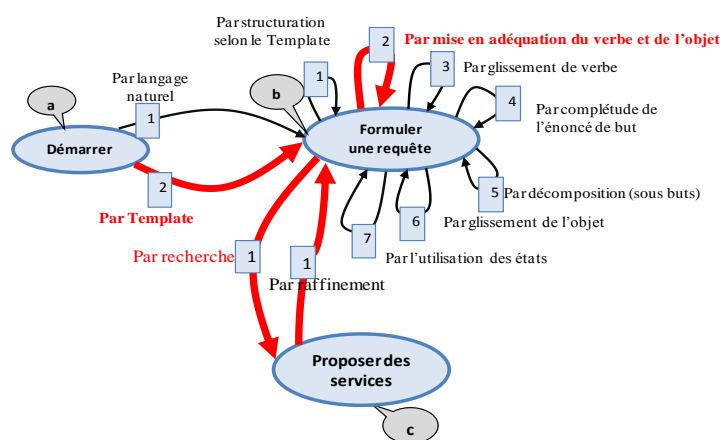


Figure 6.12. Premier Scénario de recherche de services

En utilisant la directive DRI₆, nous nous assurons que le verbe et l'objet sont compatibles (Figure 6.12), où nous examinons le lien verbe-produit dans l'ontologie de santé. Nous prenons le produit « *Personne* », et nous cherchons s'il y a une relation définie dans l'ontologie de santé entre ce produit et la classe de sens du verbe « *accueil* ». Dans cette ontologie, nous avons ajouté un lien « *personneHasRecevoir* ». Ce lien concernant les verbes que ce produit accepte. Ensuite, nous lançons la recherche des services par faire coïncidence du verbe et/ou de l'objet en suivant la directive DRI₆, mais le résultat obtenu est vide. Pour cette raison, nous utilisons la directive DRI₃ pour reformuler la requête.

4.1.1.2. Reformulation de requête

L'utilisateur peut commencer la reformulation après avoir recherché des services sans résultat. La directive DSS₂ aide dans le choix de la directive DRI à appliquer.

Dans ce cas, nous utilisons les directives DRI₇ et DRI₁₀ (en rouge gras, Figure 6.13) pour reformuler la requête, les autres directives ne sont pas utilisées puisque la requête n'est pas décomposable et le requêteur ne souhaite pas compléter son énoncé.

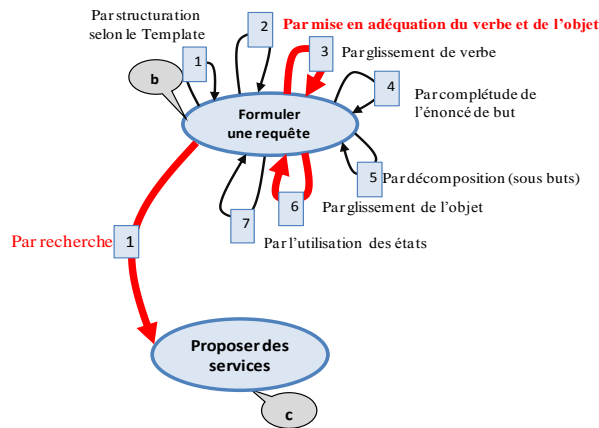


Figure 6.13. Directives utilisées dans la reformulation

4.1.1.2.1. Formulation de requête par glissement de verbe

Le verbe « *Recevoir* » appartient à la classe de sens de verbe de « *communication* » qui contient les verbes « *Recevoir* », « *Accueillir* »...

Nous pouvons reformuler en utilisant les verbes de la même classe de sens de verbe :

(*Recevoir*)_{verbe} (*une personne*)_{objet} à (*la clinique*)_{Location}.

(*Accueillir*)_{verbe} (*une personne*)_{objet} à (*la clinique*)_{Location}.

4.1.1.2.2. Formulation de requête par glissement d'objet

L'objet « *Personne* » est une généralisation de « *Patient* », « *Infirmière* », « *Employé* », « *Docteur* »...

Nous pouvons reformuler en utilisant les verbes de la même classe de sens de verbe :

(*Recevoir*)_{verbe} (*un Patient*)_{objet} à (*la clinique*)_{Location}.

(*Accueillir*)_{verbe} (*un Patient*)_{objet} à (*la clinique*)_{Location}.

(*Recevoir*)_{verbe} (*une Infirmière*)_{objet} à (*la clinique*)_{Location}.

(Accueillir) *verbe* (une Infirmière) *objet* à (la clinique) *Location*.
 (Recevoir) *verbe* (un Employé) *objet* à (la clinique) *Location*.
 (Accueillir) *verbe* (un Employé) *objet* à (la clinique) *Location*.
 (Recevoir) *verbe* (un Docteur) *objet* à (la clinique) *Location*.
 (Accueillir) *verbe* (un Docteur) *objet* à (la clinique) *Location*.

4.1.1.2.3. Recherche de services

La DRI₃ vise à rechercher des services qui répondent aux reformulations obtenues de l'application des deux directives DRI₇ et DRI₁₀.

Le résultat obtenu par faire coïncidence du verbe et de l'objet dans notre exemple est le service $S_{\text{Accueillir patient}}$ qui répond à notre formulation (Accueillir) *verbe* (un Patient) *objet* à (la clinique) *Location* à 90% (pour $\alpha = 0.9$, $\beta = 0.8$). Où le verbe « accueillir » appartient à la même classe de sens de verbe que « recevoir » (synonymie à 100%) et le produit « patient » est proche du produit « personne » à 80%. L'agrégation des deux résultats indique que le service $S_{\text{Accueillir patient}}$ satisfait la requête initiale à 72%.

4.1.1.3. Proposition des services

Après avoir trouvé des services qui satisfont la requête de l'utilisateur, on peut proposer à l'utilisateur d'explorer les composants d'un service composite, les alternatives d'un service à variation ou les services composites dont le service trouvé est un composant. La directive DSS₃ aide dans le choix de la directive DRI à appliquer.

4.1.1.3.1. Proposition des services par exploration des variantes

Dans notre exemple, le service $S_{\text{Accueillir patient}}$ est un service à variation, c'est un choix alternatif de trois services. L'application de la directive DRI₁₄ propose ces trois services :

- $S_{\text{Accueillir patient par rendez-vous}}$
- $S_{\text{Accueillir patient par planification périodique}}$
- $S_{\text{Accueillir patient par planification stratégique}}$

4.1.1.3.2. Proposition des services par exploration des composites

Dans notre exemple, le service $S_{\text{Accueillir patient}}$ est un service composant d'un service composite. L'application de la directive DRI₁₃ propose ce service :

- $S_{\text{Traiter un patient à la clinique}}$

Le processus de recherche a permis de proposer un service que l'utilisateur n'a pas prévu. En partant d'une requête « recevoir une personne à la clinique », le processus de la recherche rend un service $S_{\text{Traiter un patient à la clinique}}$. Ce résultat satisfait le but initial de l'utilisateur et répond à son besoin métier.

4.1.2. Deuxième scénario

Dans ce scénario, nous prenons l'exemple de la requête suivante : « Passer un examen médical à la clinique en payant les frais par un assureur ».

4.1.2.1. Formulation de requête par template

La requête « *Passer un examen médical à la clinique en payant les frais par un assureur* » est formalisée en utilisant la directive DRI₂, comme :

(*Passer*)_{verbe} (*un examen médical*)_{objet} en (*payant l'acte médical par un assureur*)_{manière}.

En utilisant les mêmes directives de la Figure 6.11, la directive DRI₆ assure que le verbe et l'objet sont compatibles. Nous prenons le produit « *Examen médical* », et nous cherchons s'il y a une relation définie dans l'ontologie de santé entre ce produit et la classe de sens du verbe « *investigation* ». Dans cette ontologie, nous avons ajouté un lien « *ActeMedicalHasPasser* » ; ce lien concernant les verbes que ce produit accepte. Ensuite, nous lançons la recherche des services par coïncidence du verbe et/ou de l'objet en suivant la directive DRI₆, mais le résultat obtenu est vide. Pour cette raison, nous utilisons la directive DRI₃ pour reformuler la requête.

4.1.2.2. Reformulation de requête

Dans ce cas, nous utilisons les directives DRI₇, DRI₁₀ et DRI₉ pour reformuler la requête, les autres directives ne sont pas utilisées.

4.1.2.2.1. Formulation de requête par décomposition de but

Le but (*Passer*)_{verbe} (*un examen médical*)_{objet} en (*payant les frais par un assureur*)_{manière} est décomposé en deux sous buts puisque la manière est un but, en utilisant la directive DRI₉, nous obtenons le résultat suivant :

(*Passer*)_{verbe} (*un examen médical*)_{objet} à (*la clinique*)_{Location}.

(*Payer*)_{verbe} (*l'acte médical*)_{objet} par (*un assureur*)_{manière}.

4.1.2.2.2. Formulation de requête par glissement de verbe

Le verbe « *Passer* » appartient à la classe de sens de verbe de « *investigation* » qui contient les verbes « *subir* », « *effectuer* »...

Nous pouvons reformuler la requête en utilisant les verbes de la même classe de sens de verbe de la façon suivante :

(*Subir*)_{verbe} (*un examen médical*)_{objet} à (*la clinique*)_{Location}

(*Effectuer*)_{verbe} (*un examen médical*)_{objet} à (*la clinique*)_{Location}

Ainsi, le verbe « *Payer* » appartient à la classe de sens de verbe de « *Donner de l'argent en échange de* » qui contient les verbes « *Régler* », « *Facturer* », « *Rembourser* »...

Nous pouvons reformuler en utilisant les verbes de la même classe de sens de verbe :

(*Régler*)_{verbe} un (*acte médical*)_{objet} par (*un assureur*)_{manière}.

(*Facturer*)_{verbe} un (*acte médical*)_{objet} par (*un assureur*)_{manière}.

(*Rembourser*)_{verbe} un (*acte médical*)_{objet} par (*un assureur*)_{manière}.

4.1.2.2.3. Formulation de requête par glissement d'objet

L'objet « *examen médical* » est une généralisation de « *diagnostic* », « *traitement* »..., une spécialisation de « *acte médical* » et proche de « *Maladie* », « *Symptôme* »...

Nous pouvons reformuler le premier but en utilisant le même verbe :

(Passer) *verbe* (un traitement) *objet* à (la clinique) *Location*

(Passer) *verbe* (un diagnostic) *objet* à (la clinique) *Location*.

(Passer) *verbe* (un acte médicale) *objet* à (la clinique) *Location*.

Nous pouvons reformuler le deuxième but en utilisant le même verbe :

(Payer) *verbe* un (examen médical) *objet* par (un assureur) *manière*.

(Payer) *verbe* un (acte médical) *objet* par (un assureur) *manière*.

4.1.2.2.4. Recherche de services

La DRI₃ vise à rechercher des services qui répondent aux reformulations obtenues de l'application des deux directives.

Le résultat obtenu par coïncidence du verbe et de l'objet dans notre exemple comporte les services :

- $S_{prescrire\ un\ traitement\ d'un\ patient}$ qui répond à notre formulation (*prescrire*) *verbe* (un traitement) *objet* (d'un patient) *Bénéficiaire*. Où le verbe « Prescrire » n'appartient pas à la même classe de sens de verbe que « Passer » et le produit « Examen médical » est proche du produit « Traitement » à 80%. L'agrégation des deux résultats indique que le service $S_{prescrire\ un\ traitement\ d'un\ patient}$ satisfait la requête initiale à 32% seulement (pour $\alpha = 0.9, \beta = 0.8$).
- $S_{Facturer\ un\ acte\ médical}$ qui répond à notre formulation (*Payer*) *verbe* (un acte médical) *objet* par (un assureur) *Location* à 90% (pour $\alpha = 0.9, \beta = 0.8$). Où le verbe « Payer » appartient à la même classe de sens de verbe que « Facturer » (synonymie à 100%) et le produit « acte médical » est le même. L'agrégation des deux résultats indique que le service $S_{Facturer\ un\ acte\ médical}$ satisfait la requête initiale à 80%.

Dans la suite, nous discutons les deux cas : le premier quand l'utilisateur choisit le service $S_{Diagnostiquer\ une\ maladie}$, et le deuxième quand l'utilisateur choisit le service $S_{Facturer\ un\ acte\ médical}$. Dans chacun de ces deux cas, nous montrons le déroulement de l'exploration de services.

4.1.2.3. Proposition des services

Après avoir trouvé des services qui satisfont la requête de l'utilisateur, on peut proposer à l'utilisateur d'explorer les composants d'un service composite, les alternatives d'un service à variation ou les services composites dont les services trouvés sont des composants. La directive DSS₃ aide dans le choix de la directive DRI à appliquer.

4.1.2.3.1. Proposition des services par exploration des composants

Dans notre exemple, le service $S_{prescrire\ un\ traitement}$ est un service à choix multiple composé de trois services. L'application de la directive DRI₁₅ propose ces cinq services :

- $S_{prescrire\ un\ traitement\ par\ une\ consultation\ du\ dossier\ médical}$
- $S_{prescrire\ un\ traitement\ par\ un\ acte\ chirurgical}$
- $S_{prescrire\ un\ traitement\ par\ la\ prescription\ des\ médicaments}$

Le service $S_{Facturer\ un\ acte\ médical}$ est un service composite, composé de deux services. L'application de la directive DRI₁₅ propose ces deux services :

- *Le premier est un service à choix alternatif* $S_{Préparer\ la\ facture\ de\ l'acte\ médical}$
- *Le deuxième est un service composite séquentiel* $S_{Régler\ la\ facture\ de\ l'acte\ médical}$

En appliquant la même directive sur le service $S_{\text{Régler la facture de l'acte médical}}$ qui est un service composite, composé de trois services :

- *Le premier est un service à choix multiple $S_{\text{Editer la facture de l'acte médical}}$*
- *Le deuxième est un service à choix alternatif $S_{\text{Régler une facture}}$*
- *Le troisième est un service atomique $S_{\text{Arrêter par clôture de dossier}}$*

4.1.2.3.2. Proposition des services par exploration des variantes

Dans notre exemple, le service $S_{\text{Préparer la facture de l'acte médical}}$ est un service à choix alternatif composé de deux services. L'application de la directive DRI_{14} propose ces deux services :

- $S_{\text{éditer la facture de l'acte médical par calcul automatique de frais}}$
- $S_{\text{éditer la facture de l'acte médical par demande explicite}}$

En appliquant la même directive sur le service $S_{\text{Editer la facture de l'acte médical}}$ qui est un service à choix multiple, composé de deux services. L'application de la directive DRI_{14} propose ces deux services :

- *Le premier est un service atomique $S_{\text{Editer la facture de l'acte médical avec prise en charge par un tiers}}$*
- *Le deuxième est un service atomique $S_{\text{Editer la facture de l'acte médical avec prise en charge par le patient}}$*

En appliquant la même directive sur le service $S_{\text{Régler une facture}}$ qui est un service à choix alternatif, composé de cinq services. L'application de la directive DRI_{14} propose ces cinq services atomiques :

- $S_{\text{Régler une facture par prise en charge complète}}$
- $S_{\text{Régler une facture par virement}}$
- $S_{\text{Régler une facture par carte bancaire}}$
- $S_{\text{Régler une facture par chèque}}$
- $S_{\text{Régler une facture par espèce}}$

4.1.2.3.3. Proposition des services par exploration des composites

Dans notre exemple, les services trouvés sont des services composants d'un service composite. L'application de la directive DRI_{13} propose ce service :

- $S_{\text{Réaliser une visite médicale d'un patient}}$

4.1.2.3.4. Sélection des services

Après que l'utilisateur ait exploré toutes les compositions des services, il peut sélectionner à chaque moment le/les service(s) qui répond(ent) à sa requête.

Dans notre exemple, les services qui semblent intéressants pour l'utilisateur sont :

- $S_{\text{prescrire un traitement}}$
- $S_{\text{Editer la facture de l'acte médical avec prise en charge par un tiers}}$
- $S_{\text{Régler une facture par prise en charge complète}}$

Ces trois services donnent à l'utilisateur une possibilité de répondre à ce qu'il recherche, toutefois, il peut à tout moment choisir le service $S_{\text{Traiter un patient dans la clinique}}$ qui représente tout le processus de la clinique.

5. DISCUSSION DES HYPOTHESES

Dans cette section, nous discutons les hypothèses que nous avons proposées dans l'introduction. Nous commençons par un rappel des ces hypothèses, puis un commentaire pour chacune d'elles.

5.1. Rappel des hypothèses

Dans l'introduction, nous avons formulé les trois hypothèses suivantes :

Hypothèse 1 :

La modélisation du processus de publication de services intentionnels aboutit à un guidage utile pour le fournisseur de service.

Hypothèse 2 :

La modélisation du processus de requête de services intentionnels aboutit à un guidage utile pour l'utilisateur de service.

Hypothèse 3 :

Les liens entre les ontologies de verbes et l'ontologie de produits enrichissent le processus de recherche et améliorent le résultat des requêtes.

5.2. Discussion

Les trois hypothèses sont testées grâce à une illustration de l'approche *PASIS* sur l'exemple issu du domaine de la santé que nous avons introduit dans les sections précédentes.

Hypothèse 1 :

Nous avons montré que la publication des services intentionnels est un processus. Nous avons modélisé ce processus en utilisant le modèle de la Carte. Ce processus aide le fournisseur dans sa démarche de publication par la complétude et la résolution de conflits dans l'annotation. Les annotations sont vérifiées syntaxiquement pour être conforme au modèle MiS et sémantiquement par rapport aux ontologies (vOnto, pOnto).

Le processus de guidage de la publication et le modèle MiS aident le fournisseur à décrire au plus près du métier la logique du service. De plus, la nature agrégée de ces services est supportée dans notre approche de publication puisque le modèle MiS et le descripteur intentionnel de service implémenté dans l'annuaire permettent de décrire des agrégations de services dirigées par les buts. Cette agrégation est exploitée par la publication et la recherche de services.

Hypothèse 2 :

Nous avons montré que la recherche des services intentionnels est un processus. Nous avons modélisé ce processus en utilisant le modèle de la carte. Ce processus aide l'utilisateur dans la recherche des services intentionnels qui répondent à ses besoins métier.

Nous avons montré que la recherche dans *PASIS* commence par un besoin utilisateur exprimé par un but. Ce but est l'expression la plus proche du contexte métier de l'utilisateur. Ensuite, nous avons montré comment d'autres formulations du besoin initial sont calculées. Cette requête est formulée et reformulée intelligemment à l'extérieur de l'annuaire pour tirer profit des connaissances ontologiques. La distance sémantique entre la reformulation proposée et le

besoin initial est basée sur les liens sémantiques entre les concepts de l'ontologie (*pOnto*) et est aussi calculée par le processus de reformulation. Cette distance nous a permis de délimiter le nombre de formulations.

La nature interactive du processus de recherche avec l'utilisateur permet de proposer à la fois un mode de type recherche et un mode de type exploration où d'autres choix, auxquels il n'avait pas pensé, lui sont fournis. Ceci a été montré dans le deuxième scénario de la recherche par l'utilisation des directives DRI₁₃, DRI₁₄ et DRI₁₅.

Hypothèse 3 :

Les processus de publication et de recherche de services utilisent très largement les ontologies pour capturer, du côté fournisseur, la sémantique du descripteur de service intentionnel et du côté utilisateur, le besoin à satisfaire. Les trois ontologies utilisées sont décrites dans cette thèse : *iSOnto* (ontologie technique des services intentionnels), *vOnto* (ontologie générique des verbes) et *pOnto* (ontologie de domaine des produits).

Les liens entre les ontologies de verbes et l'ontologie de produits ont permis l'enrichissement du processus de recherche. Grâce à ces liens, nous avons pu filtrer le résultat de la reformulation par la proposition des formulations dont le but contient des concepts liés seulement. Ceci a abouti à améliorer le résultat des requêtes et diminuer le champ de sélection de services trouvés pour l'utilisateur.

A la lumière de ce cas d'étude, nous avons montré que les hypothèses sont valides. Mais, une étude empirique plus profonde est nécessaire pour s'assurer de la validité de ces hypothèses. Finalement, l'implémentation de notre approche peut aider à le valider d'une manière plus sure.

6. PROTOTYPE EXPERIMENTAL

L'objectif de ce prototype est d'illustrer l'approche à travers un montage technique pour démontrer la faisabilité

6.1. Architecture du prototype

L'architecture est en deux parties (Figure 6.14) :

- Génération des descripteurs (XML et/ou BD) à partir d'une carte
 - *Technologie MetaCASE pour la construction d'un éditeur de modèle de carte et la génération de code XML (ou SQL)*
 - *Utilisation d'ontologies existantes (e-health.owl de [Klusch et al., 2006])*
- Interface simplifiée pour la formulation des requêtes et la recherche dans la base d'i-services

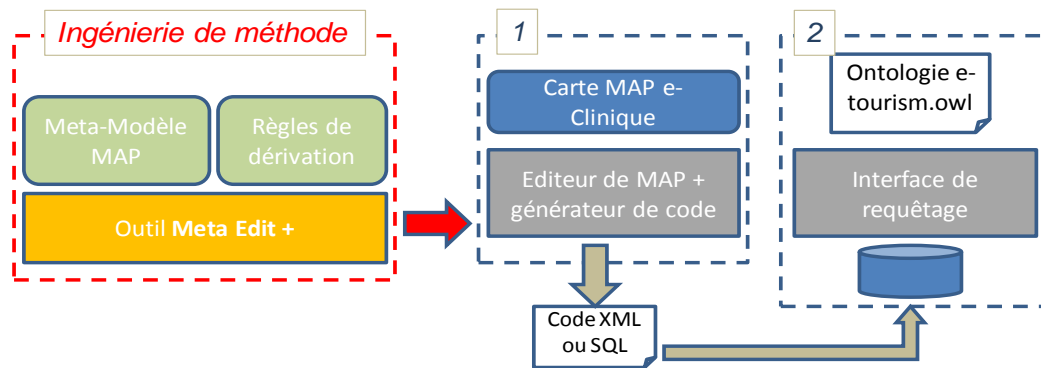


Figure 6.14. Architecture de prototype de *PASiS*

L'architecture supporte quatre parties principales :

- **Publication de services intentionnels** : Consiste à définir de nouveaux services intentionnels à l'aide d'un éditeur de MAP et d'un générateur de code ; cet éditeur permet de produire les annotations à stocker aux parties interfaces des descripteurs. Les nouveaux services sont ensuite mémorisés dans la base de services. Cette fonctionnalité aide le fournisseur à exprimer l'intention d'un service sous forme de but en utilisant le formalisme de but.
- **Reformulation de la requête** : est utilisé pour formuler les besoins de l'utilisateur ; ces besoins sont formulés sous forme de requête en utilisant le formalisme de but ; cette requête est définie et reformulée à l'aide d'un éditeur de requête.
- **Recherche d'un service intentionnel** : est utilisé pour consulter la base de services afin de trouver un service qui répond à un besoin précis de l'utilisateur. Il utilise la *Reformulation de la requête* pour aider l'utilisateur à exprimer un besoin sous forme d'une requête.
- **Proposition, exploration et sélection de services intentionnels** : est utilisé pour proposer les services résultants de la recherche, explorer les services composites et les composants afin de sélectionner un service qui répond au besoin de l'utilisateur.

Les fonctionnalités du prototype sont mises en œuvre par deux acteurs principaux :

- **Le fournisseur** de services intentionnels utilise le prototype pour la publication de services intentionnels. Il peut directement ajouter de nouveaux services intentionnels à partir d'un éditeur e MAP.
- **L'utilisateur métier** utilise le prototype pour la formulation et la reformulation de sa requête à la recherche de services intentionnels. Durant la recherche de services intentionnels, l'utilisateur intervient pour affiner la reformulation ou explorer le résultat en pour sélectionner les services qui satisfont son besoin.

Dans l'état actuel, le prototype réalise les fonctionnalités de la publication d'un service intentionnel dans une base de services, de la recherche d'un service intentionnel dans la base de services et de la reformulation de la requête de l'utilisateur.

6.2. Prototype de *PASiS*

Le prototype a été élaboré à l'aide d'une architecture client/serveur. Cette architecture comporte trois niveaux, Figure 6.15 : un niveau servant d'interface et de présentation, un

niveau pour le traitement et le contrôle et un niveau pour le stockage. Dans cette architecture client/serveur, le premier niveau est assimilé au côté client et les deux autres au côté serveur.

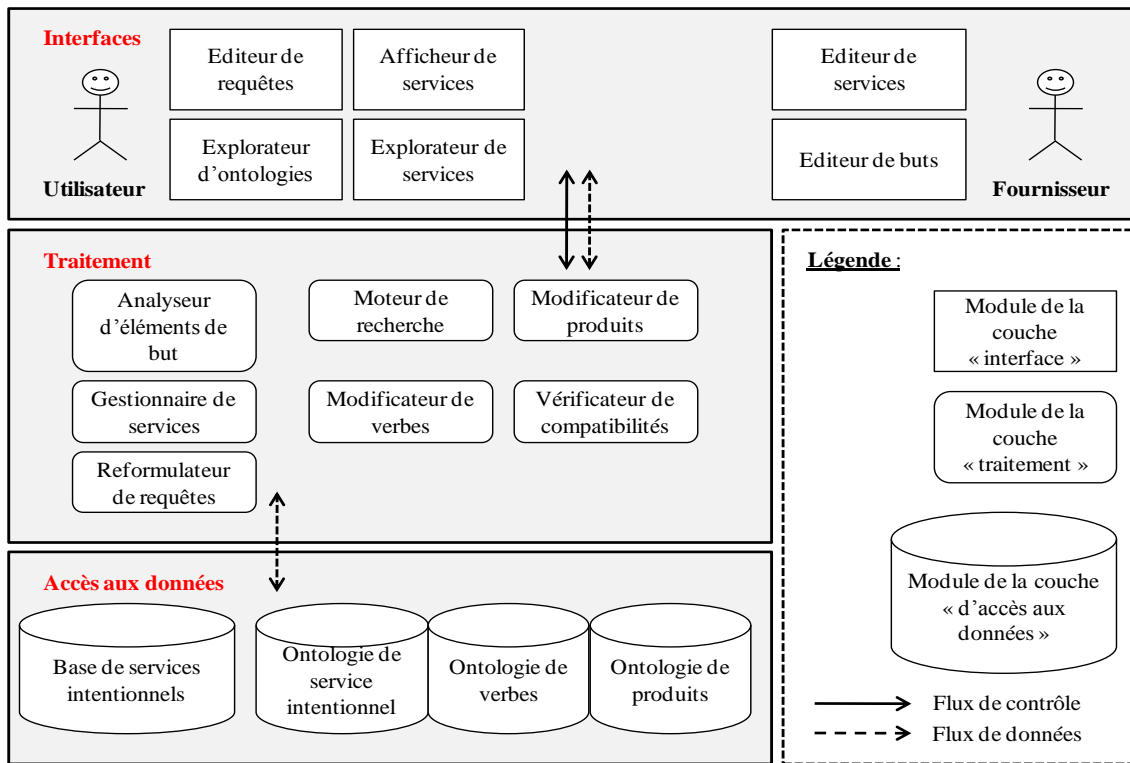


Figure 6.15. L'architecture 3-tiers du prototype *PASiS*

6.2.1. Le niveau « interface ».

Il offre un ensemble d'outils permettant aux utilisateurs de travailler avec *PASiS*. Ces outils s'adressent aux fournisseurs de services intentionnels et aux utilisateurs de services. Ils sont de trois types :

Les éditeurs permettent à travers des formulaires de saisir des requêtes et des services.

Les afficheurs permettent l'affichage des résultats ; ils permettent en particulier d'afficher la reformulation et les services intentionnels résultants.

Les explorateurs offrent la possibilité de consulter simplement la base de services et les ontologies.

6.2.2. Le niveau « traitement ».

Ce niveau offre un ensemble de modules pour formuler les requêtes, publier et rechercher les services intentionnels. Ces modules sont réalisés à l'aide de fonctions implémentées en Java. La base de données MySQL est utilisée pour la publication des descripteurs de services

intentionnels. Le niveau traitement est composé de huit modules, chaque module réalise une ou plusieurs fonctions :

- Le module « **Moteur de recherche** ». Il permet de rechercher un ou plusieurs services qui répondent à une requête (ou un but) donnée. Ce module comporte la fonction Recherche(r: requête).
- Le module « **Reformulateur de requêtes** ». Il permet d'appeler les fonctions de l'analyseur de correspondance, du gestionnaire d'ontologies et du gestionnaire de services pour déterminer quels sont les termes communs à la requête, aux ontologies et au but des services afin de trouver les reformulations qui correspondent au besoin exprimé par l'utilisateur.
- Le module « **Analyseur de but** ». Il permet d'effectuer une analyse de la requête (ou du but) en recherchant des correspondances entre le but et les ontologies, plus précisément, entre les termes définis dans les ontologies, les termes du but des services.
- Le module « **Gestionnaire de services** ». Il permet d'ajouter, de consulter et de sélectionner des services intentionnels dans la base de services.
- Le module « **Modificateur de verbes** ». Il permet de glisser et de remplacer les verbes dans le but de services intentionnels/requêtes à partir de l'ontologie de verbes.
- Le module « **Modificateur de produits** ». Il permet de proposer et de remplacer les produits dans le but de services intentionnels/requêtes à partir de l'ontologie de produits.
- Le module « **Vérificateur de compatibilité** ». Il permet vérifier la compatibilité entre le verbe et le produit dans le but de services intentionnels/requêtes à l'aide de l'ontologie de produits.

6.2.3. Le niveau « accès aux données ».

Ce niveau correspond aux données stockées qui peuvent être consultées ou mises à jour par le niveau « traitement ». Les données correspondent à la base de services intentionnels et aux trois ontologies (*iSonto*, *vOnto*, *pOnto*). Les services intentionnels sont stockés dans des tables et les ontologies sont implémentées en OWL. La section 4 présente le niveau « d'accès aux données ».

6.3. La base de services intentionnels et les ontologies

Cette section présente les principaux objets persistants de l'architecture logicielle de *PASiS* : les services intentionnels et les ontologies.

6.3.1. Les services intentionnels

Ils sont décrits au sein d'un document textuel au format XML. La structure de ce document XML est définie à l'aide du langage SAWSDL (Semantic Annotations for WSDL) [SAWSDL Working Group, 2007]. Mais, dans notre prototype, nous stockons les services intentionnels dans des tables pour des raisons de simplicité (sachant que la transformation en SAWSDL est facile).

6.3.2. Les ontologies

Les trois ontologies de *PASIS* sont décrites en OWL [Lacot, 2005]. Par exemple, l'ontologie de services intentionnels *iSonto* est partiellement définie dans la Figure 6.16. L'implémentation complète des trois ontologies est détaillée dans l'annexe B. Les trois ontologies ont été validées avec l'outil *WonderWeb OWL Ontology Validator* [Bechhofer et Volz, 2004].

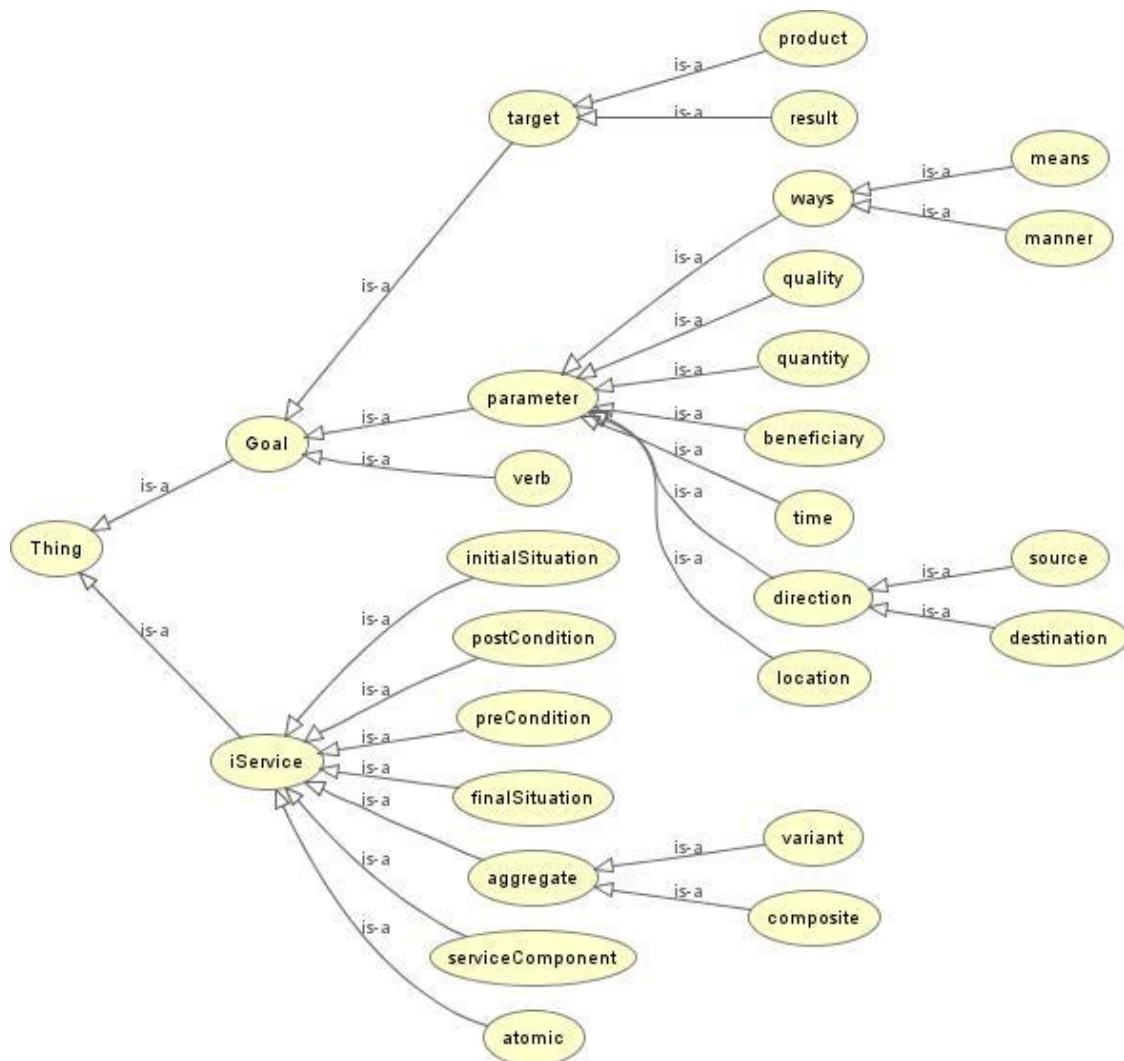


Figure 6.16. *iSonto* par Protégé

La Figure 6.17 présente l'ontologie de verbes.



Figure 6.17. Aperçu de l'ontologie de verbes

6.4. Illustration du prototype

Cette section illustre les fonctionnalités de *PASiS* à l'aide de captures d'écran du prototype. Actuellement, il offre les fonctions suivantes :

- La reformulation : aider l'utilisateur à reformuler sa requête.
- La publication : enregistrer un service intentionnel dans la base de services.
- La recherche : trouver des services intentionnels à partir d'une requête donnée.

La figure 6.18 est une capture d'écran depuis l'éditeur de carte construit avec le générateur d'outil Meta-Edit. Cet éditeur graphique repose sur la définition dans Meta-Edit de méta-modèle de la Carte. Il est généré automatiquement à partir de cette définition. Une carte est définie comme un graphe, et les concepts d'intention et de stratégie sont définis comme des objets du méta-méta-modèle GOPRR¹⁸, et le lien entre une intention et une stratégie comme des relations et des rôles.

¹⁸ GOPRR: Graph Object Property Relationship Role

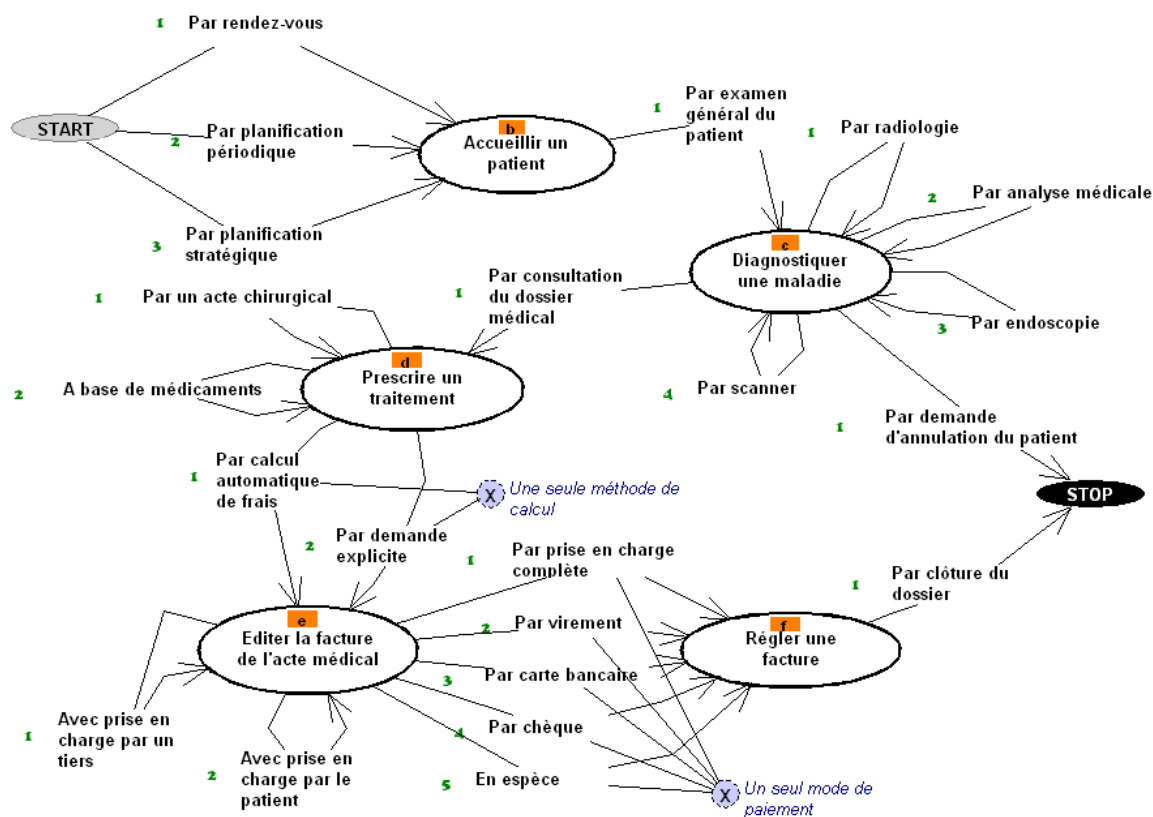


Figure 6.18. La carte du cas e-clinique telle qu'elle est définie dans l'éditeur de Meta-Edit

La capture d'écran (cf. Figure 6.19) illustre l'activité de la publication de services intentionnels. Cette capture d'écran se présente sous la forme d'un formulaire ; ce formulaire permet à un fournisseur de remplir les champs (en cochant aussi les cases) correspondants aux éléments d'un service.

Figure 6.19. *Formulaire de publication d'un service intentionnel*

Dans la capture d'écran (cf. Figure 6.20), nous montrons l'onglet concernant « service spécification », cet onglet illustre comment un fournisseur publie un service intentionnel agrégat constitué d'autres services agrégats ou atomiques.

Figure 6.20. *Formulaire de publication de spécification d'un service intentionnel*

La deuxième capture d'écran de la Figure 6.21 illustre l'éditeur de requête du prototype *PASIS*. Cet éditeur se présente sous la forme d'un formulaire. Il permet à un utilisateur

d'effectuer une recherche par le but du service intentionnel (verbe, objet, paramètres...). La recherche par le but du service analyse le verbe et l'objet contenus dans le but. Cette analyse est effectuée à la fois de manière syntaxique et en recherchant des correspondances avec l'ontologie de verbes et l'ontologie de produits.

Figure 6.21. Formulaire de recherche d'un service intentionnel

La troisième capture d'écran (cf. Figure 6.22) illustre l'affichage du résultat de la recherche des services intentionnels obtenu pour la requête « *reserve a car* ». L'affichage est composé de deux parties : une partie qui présente les services trouvés et le degré d'appariement pour chaque service trouvé et une partie qui représente l'image du concept recherché avec les produits adjacents dans l'ontologie de produits.

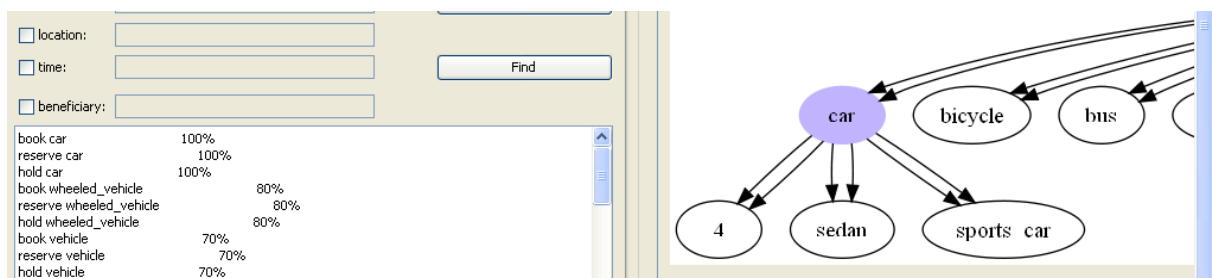


Figure 6.22. Formulaire de recherche d'un service intentionnel

7. CONCLUSION

Dans ce chapitre, nous avons présenté un cas d'application qui illustre l'utilisation des deux processus de guidage, ainsi qu'une courte discussion sur la validité des hypothèses proposées dans le chapitre de l'introduction de cette thèse.

Nous avons introduit un prototype qui permet de mettre en œuvre notre approche *PASIS*. Ce prototype permet de tester la publication, la recherche et la reformulation des requêtes des utilisateurs à la recherche des services intentionnels.

CHAPITRE 7

CONCLUSION ET PERSPECTIVES

Une difficulté fondamentale associée à l'ingénierie à base de service est celle de la discordance de niveau entre l'univers métier de l'utilisateur et celui de l'univers technique dans lequel les services sont décrits, stockés et exécutés. L'approche d'ingénierie à base de services intentionnels et le modèle MiS constituent un premier pas pour combler cette discordance et rapprocher les deux univers. Cette approche associe un but métier à un service intentionnel, et définit un processus méthodologique pour dériver ces services à partir d'une représentation à base de cartes intentionnelles du système d'information.

Cette thèse a abordé la problématique de la pertinence d'iSOA en tant qu'approche globale pour l'ingénierie à base de services orientés but. La résolution de cette problématique est faite en complétant iSOA avec des mécanismes et des processus pour la publication et la recherche de services intentionnels dans un annuaire étendu. L'approche *PASiS* proposée dans cette thèse développe un raisonnement intentionnel autour du modèle MiS qui exploite sa richesse sémantique orienté but :

- ✓ Pour accompagner le modèle MiS et l'orientation intentionnelle de sa sémantique, nous avons adopté la technique d'annotation étendue YASA pour la spécification du descripteur d'un service intentionnel. Grâce à l'ontologie technique *iSOnto* que nous avons introduit, les annotations sémantiques utilisées dans le descripteur d'un service prennent en compte la structure linguistique du modèle de but qui est sous-jacente à MiS et permettent de décrire fidèlement un service intentionnel atomique ou agrégat.
- ✓ L'approche *PASiS* présente un processus guidé lors de la publication des services intentionnels. Ce processus exploite la structure linguistique du but ainsi que la connaissance disponible dans les ontologies (domaine, technique et des verbes) pour assister le fournisseur dans la traduction des buts métiers en un réseau d'informations

sémantiques qui décrit le service intentionnel qui va être publié. Ce processus prend en considération les agrégations de services intentionnels telles que définies dans le modèle MiS.

- ✓ Le processus de recherche propose le guidage pour d'une part, aider l'utilisateur à tirer profits de toutes les possibilités qu'offre le modèle de requête, et d'autre part pour l'accompagner dans sa recherche et l'aider à trouver ce qu'il souhaite. La structure linguistique du but et la connaissance ontologique sont exploitées pour élargir et reformuler la requête initiale de l'utilisateur. Cet élargissement permet de raisonner sur la signification d'un but et d'introduire ainsi intelligence et guidage dans le processus de recherche. Enfin, et en exploitant les concepts de services intentionnels agrégés, l'utilisateur peut parcourir l'arborescence de services composites et à variation et explorer ainsi l'espace des services intentionnels disponibles.

Le recours à la modélisation avec le modèle de Carte est une caractéristique importante de cette approche. Le modèle de Carte est utilisé pour décrire le modèle de processus de la publication (MAP-Publication) et le modèle de processus de la recherche (MAP-Recherche). Les intentions et les stratégies sont documentées par des directives qui permettent de naviguer dans les cartes, et qui indiquent comment réaliser chaque intention par les stratégies qui leur sont associées. Chaque carte facilite l'accès progressif à la démarche et donne une grande liberté de choix aux utilisateurs/fournisseurs qui l'exécutent. Le modèle de Carte fournit un guidage qui permet de progresser dans le processus et de réaliser les différentes options proposées.

L'approche *PASiS* a été illustrée avec une étude de cas. Cette étude de cas est issue du domaine de santé, elle concerne la gestion des activités d'une clinique à travers la réception d'un patient, le diagnostic d'une maladie, la prescription d'un traitement et la facturation des actes. Ce cas illustratif apporte des éléments de validation aux hypothèses. En effet, le déroulement du processus de guidage lors de la publication montre clairement comment la sémantique d'un service intentionnel est préservée et comment elle se traduit dans le descripteur sémantique annoté. Et le déroulement dans l'étude de cas du processus de guidage lors de la recherche de service fournit une bonne illustration de la richesse des mécanismes de reformulation proposés et de l'exploitation qui est faite de la connaissance disponible dans les ontologies de l'approche *PASiS*. Enfin, les liens que nous avons introduits entre les verbes et concepts de domaine permettent d'introduire de nouvelles manières d'exploration de l'univers sémantique de la recherche, et de filtrer le résultat d'une requête selon des critères de compatibilité entre les éléments composant la requête.

Finalement, nous avons présenté un prototype de l'approche *PASiS* qui a été développé pour illustrer sa faisabilité. Cette plate-forme est une juxtaposition de plusieurs composants logiciels, et offre les fonctionnalités nécessaires pour illustrer la publication des descripteurs de services intentionnels, la reformulation de requêtes des utilisateurs et la recherche de services.

PERSPECTIVES

Ce travail ouvre la voie vers différentes perspectives. Il peut être poursuivi dans plusieurs directions :

Implémentation : le prototype que nous avons évoqué a été conçu pour démontrer la faisabilité de l'approche, il ne permet pas encore une expérimentation réelle de celle-ci. Il est donc tout à fait judicieux de consolider le travail qui a été fait et de transformer ce prototype

en un outil un peu plus complet et plus fiable pour pouvoir effectuer des tests plus approfondis, notamment avec des étudiants en niveau Master recherche.

Expérimentation : la disponibilité d'un prototype telle qu'elle vient d'être mentionnée ouvre la voie à des expérimentations à échelle réelle de l'approche *PASIS*. La description d'un service qui soit aussi proche de l'univers métier de l'utilisateur reste pour le moment une proposition issue des laboratoires de recherche, et il est nécessaire de la confronter avec les usages du monde réel. Il serait ainsi extrêmement intéressant de confronter l'approche *PASIS* avec les autres approches existantes (telles que celles que nous avons étudiées dans l'état de l'art) pour mesurer l'efficacité d'une approche d'ingénierie à base de service intentionnels versus une approche d'ingénierie à base de service techniques. De telles confrontations nécessitent la mise en place de protocoles rigoureux d'expérimentations empiriques.

Développement de nouveaux cas : dans la lignée des deux perspectives qui viennent d'être mentionnées, il est nécessaire de construire de nouveaux cas et d'y appliquer l'approche *PASIS* afin d'une part, valider les deux processus proposés, et d'autre part, enrichir la base des exemples illustratifs de cette approche. Ce qui permettra aussi de construire et d'enrichir un annuaire d'i-services.

BIBLIOGRAPHIE

- [Aljoumaa, 2010a] K. Aljoumaa, “Publishing and Discovery of Intentional Services: Goal-driven Approach”, *In Proc. of the CAiSE Doctoral Consortium 2010*, CEUR-WS.org/Vol-593, Hammamet, Tunisia, 8 Juin, 2010. Disponible sur <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-593/Paper01.pdf>
- [Aljoumaa, 2010b] K. Aljoumaa, “Vers une ontologie partagée pour formuler le but de l'utilisateur et l'intention de service”, *Manifestation des Jeunes Chercheurs en Sciences et Technologies de l'Information et de la Communication (MajecSTIC'2010)*, Bordeaux, France, 13-15 October 2010.
- [Aljoumaa et al., 2010] K. Aljoumaa, S. Assar, C. Souveyet, “Publishing intentional services using new annotation for WSDL”, *In Proc. of the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS2010)*, ACM, Paris, France, p. 881-884, 08-10 November 2010.
- [Aljoumaa et al., 2011a] K. Aljoumaa, S. Assar, C. Souveyet, “Reformulating user's queries for Intentional Services Discovery using Ontology-based Approach”, *In Proc. of the fourth IFIP International Conference on New Technologies, Mobility and Security (NTMS'2011)*, Paris, France, p. 1-4, 7-10 February 2011.
- [Aljoumaa et al., 2011b] K. Aljoumaa, S. Assar, C. Souveyet, “Publishing Intentional Services using extended semantic annotation”, *In Proc. of the IEEE Fifth International Conference on Research Challenges in Information Science (RCIS 2011)*, Guadeloupe - French West Indies, France, pp. 138-146, 19-21 May 2011.
- [Aljoumaa et al., 2011c] K. Aljoumaa, S. Assar, C. Souveyet, “Publication des services intentionnels, Processus de guidage d'annotation du descripteur intentionnel”, *proposal paper in inforsid 2011*, Lille, France, 24-26 May 2011.
- [Al-Masri et al., 2008] E. Al-Masri, Q. H. Mahmoud, “Investigating Web Services on the World Wide Web”, *In Proc. of WWW 2008 conference*, Beijing, China, 21-25 Avril, 2008.
- [Alonso et al., 2004] G. Alonso, F. Casati, H. Kuno, V. Machiraju, *Web Services. Concepts, Architectures and Applications*, Springer, 2004.
- [Arni-Bloch et al., 2009] N. Arni-Bloch, J. Ralyté, L. Michel, *Service-Driven Information Systems Evolution: Handling Integrity Constraints Consistency*, Springer, 2009.
- [Arsanjani et al., 2004] A. Arsanjani, F. Casati, “Service-Oriented Modeling and Architecture”, *IBM developer works*, 2004.
- [Assar et Aljoumaa, 2011] S. Assar, K. Aljoumaa, “Matching user requirements with query formulations in intentional service oriented computing”, *In Proc. of the 35th annual IEEE International Computer Software and Applications Conference (COMPSAC 2011)*, Munich, Germany, pp. 476-481, Juillet 18-21, 2011.
- [Assar et al., 2003] S. Assar, F. Semmak, R. Barkhi, “Component-based business information systems”, *In Proc. of workshop edited by Conf. OOIS'03*, Genève, September 2003.
- [Baida et al., 2004] Z. Baida, J. Gordijn, B. Omelayenko, H. Akkermans, “A Shared Service Terminology for Online Service Provisioning”, *In Proc. of the Sixth International Conference on Electronic Commerce (ICEC04)*, Delft, Netherlands, 2004.

- [Benjamin, 1999] A. Benjamin, “Une Approche Multi-démarches pour la modélisation des démarches méthodologiques”, Thèse de doctorat en informatique de l’Université Paris 1, October 1999.
- [Bianco et al., 1999] G. Bianco, V. De Antonellis, S. Castano, M. Melchiori, “A Markov Random Field Approach for Querying and Reconciling Heterogeneous Databases”, *In Proc. Of the 10th International Workshop on Database and Expert Systems Applications (DEXA '99)*, Florence, Italy, September 1999.
- [Bouchachia et Mittermeir, 2001] A. Bouchachia, R.T. Mittermeir, “Coping with Uncertainty in Software Retrieval Systems”, *In Proc. of the 2nd Intl. Workshop on Soft Computing Applied to Software Engineering (SCASE.01)*, pp. 21-30, February 2001.
- [Bresciani et al., 2004] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, J. Mylopoulos, “Tropos: An agent-oriented software development methodology,” *Autonomous Agents and Multi-Agent Systems*, vol. 8(3), pp. 173–206, 2004.
- [Castano et al., 1992] S. Castano, V. De Antonellis, B. Zonta, “Classifying and Reusing Conceptual Components”, *In Proc. Of the 11th International Conference on Conceptual Modelling (ER '92)*, Karlsruhe, pp. 121-138, 1992.
- [Castano et De Antonellis, 1993] S. Castano De V. Antonellis, “A Constructive Approach to Reuse of Conceptual Components”, *In Proc. Of Advances in Software Reuse: Selected Papers from the Second International Workshop on Software Reusability*, Lucca, Italy, 24-26 March, 1993.
- [Chabeb et Tata, 2008] Y. Chabeb, S. Tata, “Yet Another Semantic Annotation for WSDL (YASA4WSDL)”, *in IADIS WWW/Internet 2008 Conference*, pp. 462–467, October 2008.
- [Chabeb et al., 2009] Y. Chabeb, S. Tata, D. Belaid, “Toward an integrated ontology for web services”, *In Proc. of Fourth International Conference on Internet and Web Applications and Services (ICIW '09)*, pp. 462–467, May 2009.
- [Chabeb et al., 2010] Y. Chabeb, S. Tata, A. Ozanne, “YASA-M: Semantic Web Service Matchmaker”, *International Conference on Advanced Information Networking and Applications*, Perth, Australia, 20-23 April 2010.
- [Chinnici et al., 2007] R. Chinnici, S. Weerawarana, J.J. Moreau, A. Ryman, *Web services description language (WSDL)*, version 2.0 part 1: Core language. W3C recommendations. June 2007. W3C. <http://www.w3.org/TR/2007/REC-wsdl20-20070626>.
- [Corby et al., 2006] O. Corby, C. Faron-Zucker, R. Dieng-Kuntz, F. Gandon, “Searching the semantic web: Approximate query processing based on ontologies”, *IEEE Intelligent Systems Journal*, 21(1), 2006.
- [Corby et al., 2009a] I. Mirbel. “Démarches sémantiques de recherche d'information sur le Web”, *In Proceeding Ingénierie des Connaissances IC'09*, Hammamet, Tunisia, Juin 2009.
- [Corby et al., 2009b] O. Corby, C. Faron-Zucker, I. Mirbel, “Implementation of Intention-Driven Search Processes by SPARQL Queries”, *In Proc. of Web Science WebSci'09*, Athens, Greece, March 2009.
- [Da Silva Santos et al., 2008] L. O. B. Da Silva Santos, L. F. Pires, M. van Sinderen, “A Goal-based Framework for Dynamic Service Discovery and Composition”, *International Workshop on Architectures concepts and Technologies for Service Oriented*, Porto, Portugal, p. 67-78, 2008.
- [Da Silva Santos et al., 2009] L. O. B. Da Silva Santos, G. Guizzardi, L. F. Pires, M. van Sinderen, “From User Goals to Discovery and Composition. Advances in Conceptual Modeling - Challenging Perspectives”, *Lecture Notes in Computer Science 5833/2009*. Springer Verlag, Berlin, pp. 265-274, 2009.
- [De Bruijn et al., 2005] J. De Bruijn, H. Lausen, D. Fensel, “D16.1v0.21 the Web Service Modelling Language WSML”, disponible sur <http://www.wsmo.org/TR/d16/d16.1/v0.21/>, October 2005.

- [Dik, 1989] S.C. Dik, *The theory of functional grammar*, Foris publications, Dodrecht, Netherlands, 1989.
- [Driss et al., 2010] M. Driss, N. Moha, Y. Jamoussi, J-M. Jézéquel, H. Hajjami Ben Ghézala, “A Requirement-Centric Approach to Web Service Modeling, Discovery, and Selection”, *In Proceedings of the 8th International Conference on Service Oriented Computing (ICSOC 2010)*. San Francisco, USA, December, 2010.
- [Eck et Wieringa, 2003] P. v. Eck, R. Wieringa, “Requirements Engineering for Service Oriented Computing: A Position Paper”, *In Proc. of the 1st International E-Services Workshop (ICEC 2003)*, Pittsburgh, USA, pp.23-28, 30 September, 2003.
- [Eckert et al., 2009] J. Eckert, M. Bachhuber, N. Repp, R. Steinmetz, “The Implementation of Service-Oriented Architectures in the German Banking Industry - A Case Study”, *In Proceedings of AMCIS 2009*. p. Paper 473, 2009.
- [Elbyed, 2008] A. Elbyed, *Ontologie mapping pour l’intégration des données hétérogènes*, Thèse de doctorat en informatique, TELECOM et Management SudParis, Evry, France 2008.
- [ESSI WSMO, 2004] ESSI WSMO working group, *Web Service Modeling Ontology*. <http://www.wsmo.org/>, 2004.
- [Farrell et Lausen, 2007] J. Farrell, H. Lausen, “Semantic Annotations for WSDL and XML Schema”, *W3C Recommendation* <http://www.w3.org/TR/2007/REC-sawSDL-20070828/>, 28 August 2007.
- [Fauvet et al., 2002] M-C. Fauvet, M. Dumas, B. Benatallah, “Collecting and querying distributed traces of composite service executions”, *Confederates International conferences CoopIS, DOA and ODBASE*, Springer p. 373-390, Irvine, California, USA, 2002.
- [Fensel et Bussler, 2002] D. Fensel, C. Bussler, “The Web Service Modeling Framework WSMF”, *Electronic Commerce Research and Applications*, 1(2):113–137, 2002.
- [Fillemore, 1968] C.J. Fillemore, *The case for case, in Universals in linguistic theory*, Holt, Rinehat and Winstonc inc., E.Bach/R.T.Harms (eds), 1968.
- [Fujii et Suda, 2004] K. Fujii, T. Suda, “Dynamic Service Composition Using Semantic Information”, *In Proc.. of ICSOC*, , ACM 2004, p. 39-48 New York, USA, November 2004.
- [Ganter et Wille, 1999] B. Ganter, R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag New York, Inc., Heidelberg, 1999.
- [Gomez et al., 2004] J.M. Gómez, M. Rico, F. García-Sánchez, R. Martínez-Béjar, C. Bussler, “GODO: Goal-driven orchestration for Semantic Web Services”, *In Proc. of the Workshop on Web Services Modeling Ontology Implementations (WIW 2004)*, Frankfurt, Germany, 29-30 September, 2004.
- [Gomez et al., 2006] J. Gomez, M. Rico, F. Garcia-Sanchez, “GODO: Goal Oriented Discovery for Semantic Web Services”, *5th international Semantic Web Conference*, 2006.
- [He et Chang, 2003] B. He, K.C.C. Chang, “Statistical Schema Matching across Web Query Interfaces”, *In SIGMOD Conference*, p. 217–228, 2003.
- [Hirschheim et al., 2010] R. Hirschheim, R. Welke, A. Schwarz, “Service-Oriented Architecture: Myths, Realities, and a Maturity Model”, *MIS Quarterly Executive*, 9, 37–48, 2010.
- [Issarny et al., 2007] V. Issarny, M. Caporuscio, N. Georgantas, “A Perspective on the Future of Middleware-based Software Engineering”, *In: Briand, L. and Wolf, A. (Eds.), Future of Software Engineering 2007 (FOSE), ICSE (International Conference on Software Engineering), IEEE-CS*, 2007
- [Jarke et al., 1992] M. Jarke, J. Mylopoulos, J.M. Schmidt, Y. Vassiliou, “DAIDA - An Environment for Evolving Information Systems”, *ACM Trans., In Information Systems*, vol. 10, n° 1, 1992.

- [Jarke et Pohl, 1992] M. Jarke, K. Pohl, “Requirements Engineering: An Integrated View of Representation, Process and Domain”, *In Proc. of the 4th European Software Conference*, Springer Verlag, 1993.
- [Jarke et al. 1999] M. Jarke, C. Rolland, A. Sutcliffe, R. Domges, *The NATURE requirements Engineering*. Shaker Verlag, Aachen 1999.
- [Jilani et al., 1997] L.L. Jilani, R. Mili, A. Mili, “Approximate Component Retrieval: An Academic Exercise or a Practical Concern?”, *In Proc.. Of the 8th Workshop on Institutionalising Software Reuse (WISR8)*, Columbus, Ohio, Mars 1997.
- [JointUS / UE, 2001] Joint US/EU ad hoc Committee, “Reference Description of the DAML_OIL Ontology Markup Language”, version 4.2, disponible sur <http://www.daml.org/2001/03/reference>, Mars 2001.
- [Kaabi, 2007] R. S. Kaabi, Une Approche Méthodologique pour la Modélisation Intentionnelle des Services et leur Opérationnalisation, Thèse du doctorat, Centre de Recherche en Informatique, Université de Paris 1 Panthéon-Sorbonne, février 2007.
- [Kaabi et Souveyet, 2007] R. S. Kaabi, C. Souveyet, “Capturing Intentional services with Business Process Maps”, *Research Challenges in Information Science (RCIS)*, Ouarzazate, Morocco, 2007.
- [Kaufer et Klüsch, 2006] F. Kaufer et M. Klüsch, “Hybrid Semantic Web Service Matching with WSMO-MX”, *In Proc.. 4th IEEE European Conference on Web Services (ECOWS)*, Zurich, Switzerland, IEEE, 2006.
- [Keller et al., 2005] U. Keller, R. Lara, H. Lausen, A. Polleres, et D. Fensel, “Automatic location of services”, *In Proc. of the 2nd European Semantic Web Conference (ESWC)*. Heraklion, Crete: LNCS 3532, Springer, 2005, pp. 1–16. Available: <http://www.informatik.unitrier.de/ley/db/conf/esws/eswc2005.html#KellerLLPF05>
- [Klein et al., 2005] M. Klein, B. König-Ries, M. Mussig, “What is needed for semantic service descriptions ? A proposal for suitable language constructs”, *Int. J. Web Grid Serv.*, 1(3/4):328–364, 2005.
- [Klüsch et al., 2006] M. Klüsch, B. Fries, K. Sycara, “Automated semantic web service discovery with OWLS-MX”, *In Proc. of the fifth international joint conference on Autonomous agents and multiagent systems*, ACM Press, pp. 915–922, New York, USA, 2006.
- [Klüsch et al., 2008] M. Klüsch, P. Kapahnke, F. Kaufer, “Evaluation of WSML Service Retrieval with WSMO-MX”, *in IEEE International Conference on Web Services, ICWS '08*, pp. 401–408, September 2008.
- [Klüsch et Kapahnke, 2008] M. Klüsch et P. Kapahnke, “Semantic Web Service Selection with SAWSDL-MX”, *in SMRR, ser. CEUR Workshop*, R. L. Hernandez, T. D. Noia, and I. Toma, Eds., vol. 416. CEUR-WS.org, 2008.
- [Klüsch et al., 2009] M. Klüsch, P. Kapahnke, I. Zinnikus, “SAWSDL-MX2: A Machine-Learning Approach for Integrating SemanticWeb ServiceMatchmaking Variants”, *In ICWS*, pages 335–342, 2009.
- [Kim et al., 2010] S. Kim, M. Kim, V. Sugumaran, S. Park, “A Scenario Based Approach for Service Identification” *34th Annual IEEE Computer Software and Applications Conference Workshops*, pp. 237–238, 2010.
- [Kung-Kiu et Zheng, 2007] L. Kung-Kiu, W. Zheng, “Software Component Models”, *IEEE Transactions on Software Engineering*, vol. 33, no. 10, pp. 709-724, July 2007
- [Lamsweerde et Letier, 2000] A.V. Lamsweerde, E. Letier, “Handling obstacles in goal oriented requirements engineering”, *IEEE Transactions on Software Engineering*, vol. 26, n° 10, p. 978–2000.

- [Lara et al., 2004] R. Lara, D. Roman, A. Polleres, D. Fensel, "A Conceptual Comparison of WSMO and OWL-S", *In Proc. of ECOWS 2004*, Springer, Erfurt, Germany, 27-30 September, 2004.
- [Lichtenstein et al., 2005] S. Lichtenstein, L. Nguyen, A. Hunter, "Issues in IT service-oriented requirements engineering", *Australasian journal of information systems*, vol 13, 2005.
- [Levi et Arsanjani, 2002] K. Levi, A. Arsanjani, "A goal-driven approach to enterprise component identification and specification", *Communications of the ACM*, vol 45, pp. 45-52, 2002.
- [Li et Horracks, 2003] L. Li, I. Horracks, "A Software Framework for Matchmaking based on Semantic Web Technology", *In Proc. of the World Wide Web 2003 Conference*, New York, ACM, p. 331-339, Budapest, Hungary, May 2003.
- [Mandell et McIlraith, 2003] D. Mandell, S. McIlraith, "Adapting bpel4ws for the semantic web: the bottom-up approach to web service interoperation", *In Int. Semantic Web Conference (ISWC)*, Lecture Notes in Computer Science 2870 Springer, p. 227-241, Florida, USA, September 2003.
- [Martin et al., 2004a] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, K. Sycara, "OWL-S: Semantic Markup for Web Services", In The OWL Services Coalition version 1.0, 2004. Disponible sur <http://www.daml.org/services/owl-s/1.0/owl-s.pdf>.
- [Martin et al., 2004b] D. Martin, M. Paolucci, S. Mcilraith, M. Burstein, D. Mcdermott, D. McGuinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, N. Srinivasan, K. Sycara. *Bringing Semantics to Web Services: The OWL-S Approach*. Springer, 2004.
- [Martin et Domingue, 2007a] D. Martin, J. Domnigue, "Semantic web services, Part1", *Intelligent Systems*, IEEE, 22(5), p.12-17, 2007.
- [Martin et Domingue, 2007b] D. Martin, J. Domnigue, "Semantic web services, Part2, *Intelligent Systems*, IEEE, 22(6), p.8-15, 2007.
- [McIlraith et al, 2001] S.A. McIlraith, T.C. Son, Z. Honglei, "Semantic web services", *Intelligent Systems*, IEEE, 16(2), 2001.
- [METEOR-S, 2005] METEOR-S: Semantic Web Services and Processes. Disponible sur <http://lsdis.cs.uga.edu/projects/meteor-s/index.php?page=1>, 2005.
- [Miller, 1993] K. Miller, Introduction to WordNet: an On-line Lexical Database Distributed with WordNet software, 1993.
- [Mirbel et Crescenzo, 2009] I. Mirbel, P. Crescenzo, "Des besoins des utilisateurs à la recherché de services web : une approche sémantique guidé par les intentions", Rapport de recherche ISRN I3S/RR-2009-12-FR, 2009.
- [Nurcan et Rolland, 1999] S. Nurcan et C. Rolland, "Using EKD-CMM electronic guide book for managing change in organizations", *European-Japanese Conference on Information Modeling and Knowledge Bases (ECJ)*, pp. 105-123, Iwate, Japan, 1999.
- [Oren et al., 2004] E. Oren, M. Zaremba, M. Moran, "Overview and Scope of WSMX", WSMX Working Draft, 2004. Disponible sur <http://www.wsmo.org/2004/d13/d13.0/v0.1/20040611/>.
- [OWL-S, 2003] OWL-S, "Semantic Markup for Web Services. The OWL Service Coalition", disponible sur <http://www.daml.org/services/owl-s/1.0/>, 2003.
- [Pallos, 2001] M. Pallos, *Service Oriented Architecture: A Primer*, EAI Journal, Decembre 2001.
- [Paolucci et al., 2002a] M. Paolucci, T. Kawamura, T. R. Payne, K. Sycara, "Semantic Matching of Web Services Capabilities", in *International Semantic Web Conference (ISWC)*, Sardinia, Italy, 2002.
- [Paolucci et al., 2002b] M. Paolucci, T. Kawamurea, T. Payne, and K. Sycara, "Importing the Semantic Web in UDDI", *In Proc. of Web Services, E-Business, and Semantic Web Workshop*, pp. 225-236, 2002.

- [Paolucci et al., 2003] M. Paolucci, A. Ankolekar, N. Srinivasan, K. Sycara, "The DAML-S Virtual Machine", *In Proc. of the Second International Semantic Web Conference (ISWC)*, pp. 290-305, Sandial Island, FL, USA, October 2003.
- [Papazoglou et Grefen, 2002] M. P. Papazoglou et P. Grefen, "Service-Oriented Computing", *The S.O.C Manifesto*, 2002.
- [Papazoglou et Van den Heuvel, 2006] M. P. Papazoglou, W-J. Van den Heuvel, "Service-Oriented Design and Development Methodology", *Int. J. of Web Engineering and Technology (IJWET)*, 2006.
- [Papazoglou et Van den Heuvel, 2007] M. P. Papazoglou, W-J. Van den Heuvel, "Service Oriented Architectures: Approaches, Technologies and Research Issues", *VLDB Journal*, vol. 14(3), pp. 389-413, 2007.
- [Papazoglou et al., 2008] M. P. Papazoglou, P. Traverso, S. Dustdar F. Leymann, "Service-Oriented Computing: a Research Roadmap", *International Journal of Cooperative Information System*. Vol. 17(2), pp. 193-225, 2008.
- [Patil et al., 2004] A. Patil, S. Oundhakar, A. Sheth, K. Verma , "METEOR-S Web service Annotation Framework WWW 2004", New York, New York, USA May 17-22, 2004.
- [Penserini et Mylopoulos, 2005] L. Penserini, J. Mylopoulos, "Design Matters for Semantic Web Services", *Technical report: T05-04-03*, ITC-IRST, Avril 2005.
- [Penserini et al. 2007] L. Penserini, A. Perini, A. Susi "High Variability Design for Software Agents: Extending Tropos", *ACM Transaction on autonomous And Adaptive Systems*, Vol.2, No. 4, Article 16, Novembre 2007.
- [Peppard, 2003] J. Peppard, "Managing IT as a Portfolio of Services", *European Management Journal*, 21, 467-483, 2003.
- [Perini et al., 2005] A. Perini, A. Susi, J. Mylopoulos, "Tropos Design Process for Web Services", *1st International Workshop on Service-Oriented Computing: Consequences for Engineering Requirements*, Paris, Septembre 2005.
- [Plihon, 1994] V. Plihon, The OMT Methodology, The OOA Methodology (Coad/yourdon), The SA/SD Methodology, The Entity Relationship Methodology, The O* Methodology, The OOD Methodology, NATURE Deliverable DP2, 1994.
- [Plihon et Rolland, 1995] V. Plihon, C. Rolland, Modelling Ways-of-Working, *Proc 7th Int. Conf. on Advanced Information Systems Engineering (CAISE)*, Springer Verlag (Pub.), 1995.
- [Plihon et Rolland, 1997] V. Plihon, C. Rolland, "Using a generic approach to support the construction of methods", *In Proc. of the 8th International Conference on Database and Expert Systemes Applications (DEXA'97)*, Toulouse, France, 1-7 September, 1997.
- [Ponnekanti et Fox. 2002] S. R. Ponnekanti, A. Fox, "SWORD : A Developer Toolkit for Web Service composition", *alternate paper of the Web Engineering Track of the eleventh World Wide Web Conference*, Honolulu, Hawaii, 7-11 Mai 2002, disponible sur <http://www2002.org/CDROM/alternate/786/>.
- [Prat, 1997] N. Prat, Goal formalisation and classification for requirements engineering, *In Proc. of the Third International Workshop on Requirements Engineering: Foundations of Software Quality REFSQ'97*, Barcelona, June 1997.
- [Prat, 1999] N. Prat, Réutilisation de la trace par apprentissage dans un environnement pour l'ingénierie des processus, Thèse de doctorat en informatique, Université Paris1, 1999.
- [Prieto-Diaz, 1991] R. Prieto-Diaz, "Implementing Faceted Classification for Software Reuse", *Communications of the ACM, Special issue on software engineering*, 34(5):88 - 97, 1991.

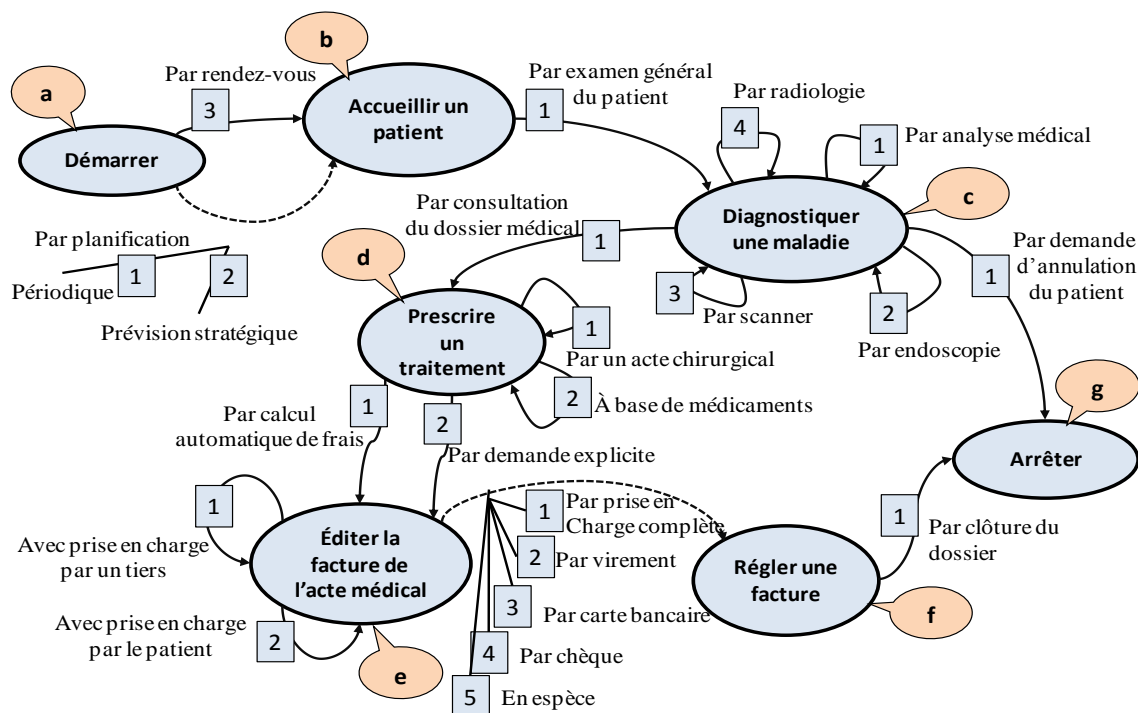
- [Psyche et al., 2003] V. Psyche, O. Mendes, J. Bourdeau, “Apport de l’ingénierie ontologique aux environnements de formation à distance”, *Revue STE*, pages 89–126. 2003.
- [Quartel et al., 2004a] D. Quartel, R-M. Dijkman, M. van Sinderen, “Methodological Support for Service-oriented Design with ISDL”, *In Proc. of the 2nd International Conference on Service Oriented Computing (ICSOC’04)*, New York, USA, 2004.
- [Ralyté, 2001] J. Ralyté, “Ingénierie des méthodes à base de composants”, Thèse de Doctorat, Université de Paris I, Janvier 2001.
- [Rolland 1997] C. Rolland, A primer for method engineering”, *proceeding of INFORSID’ 97*, Toulouse, France (1997).
- [Rolland et al., 1998] C. Rolland, C. Ben Achour, C. Cauvet, J. Ralyté, A. Sutcliffe, N-A. M. Maiden, M. Jarke, P. Haumer, K. Pohl, E. Dubois and P. Heymans, “A Proposal for a Scenario Classification Framework”, *Requirements Engineering Journal (REJ)*, Vol. 3, N°1, pp. 23-47, 1998.
- [Rolland et al., 1999] C. Rolland, N. Prakash, A. Benjamen, “A multi-model view of process modelling”, *Requirements Engineering*, 4(4), 169–187. 1999.
- [Rolland et Parkash, 2000] C. Rolland, N. Prakash, “Bridging the gap between Organizational needs and ERP functionality”, *Requirements Engineering Journal*, 2000.
- [Rolland et Kaabi, 2005] C. Rolland, R-S. Kaabi, “Designing Service Based Cooperative Systems”, *Encyclopedia of ETechnologies and Applications (EET&A)*, IDEA Group (pub), 2005
- [Rolland et al., 2007] C. Rolland, R-S. Kaabi, N. Kraeim, “On ISOA: Intentional Services Oriented Architecture”, *International Conference on Advanced information Systems Engineering (CAISE)*, Springer-Verlag, Trondheim, Norway, 11 Juin 07.
- [Rolland, 2007] C. Rolland, “Conceptual Modelling in Information Systems Engineering”, *Chapter Capturing System Intentionality with Maps*, Springer-Verlag, pp. 141 - 158, Berlin heidelberg, Germany 2007.
- [Rolland et al., 2008] C. Rolland, C. Souveyet, N. Kraiem, “An Intentional View of Service-Oriented Computing”, *Revue Ingénierie des Systèmes d’Information (ISI)*, *RSTI (Revue des Sciences et Technologies de l’Information)*, ISI 13[1]: 107–137, 2008.
- [Rolland et al., 2010] C. Rolland, M. Kirsch-Pinheiro, C. Souveyet, “An Intentional Approach to Service Engineering”, *IEEE Transactions On Services Computing*, vol. 3, n°4, pp. 292-305, (2010).
- [Roman et al., 2005] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, D. Fensel, “Web Service Modeling Ontology”, *Applied Ontology*, 1(1): 77-106, 2005.
- [Salton et McGill, 1983] G. Salton, et M. J. McGill, “Introduction to Modern Information Retrieval”, McGraw-Hill. 1983: 118 . 156.
- [Sassen et Macmillan, 2005] A. Sassen, C. Macmillan, “The service engineering area: An overview of its current state and a vision of its future”, Disponible sur ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/stds/sota_v1-0.pdf. European Commission, Network and Communication Technologies, Software Technologies, 2005.
- [Sawyer et al., 2005] P. Sawyer, J. Hutchinson, J. Walkerdine, I. Sommerville, Faceted Service Specification”, *In Proc. SOCCER (Service-Oriented Computing: Consequences for Engineering Requirements) Workshop, at RE’05 Conference*, Paris France, August 2005.
- [Semmak, 1998] F. Semmak, Réutilisation de composants de domaine dans la conception de systèmes d’Information. Thèse de doctorat de l’Université de Paris I Panthéon Sorbonne, 1998

- [Sivashanmugam et al., 2003] K. Sivashanmugam, K. Verma, A. Sheth, J. Miller, “Adding Semantics to Web Service Standards”, *In Proc. of the 1st International Conference on Web Services*, Las Vegas, NV, pp. 395-401, 2003.
- [Smith et al., 2004] M.K. Smith, C. Welty, D.L. McGuinness, “OWL Web Ontology Language Guide”, <http://www.w3.org/TR/2004/RECowlguide20040210/>, 2004.
- [Sycara et al., 2003] K. Sycara, M. Paolucci, A. Ankolekar, N. Srinivasan, “Automated discovery, interaction and composition of semantic web services”, *Journal of Web Semantics*, pages 27-46, 2003.
- [Szyperski, 1997] C. Szyperski. *Component software: beyond object-oriented programming*. New York : ACM Press Harlow, England Reading, Mass : Addison- Wesley, 1997.
- [Tansey et Stroulia, 2006] B. Tansey, E. Stroulia, *Towards an Economics Model for Software Development in Support of B2C Services*, 2006.
- [Urrego, 2005] G. S. Urrego, “Approches linguistiques pour l’ingénierie de besoins fonctionnels et non- fonctionnels”, Thèse du doctorat, Centre de Recherche en Informatique, Université de Paris 1 Panthéon-Sorbonne, 2005.
- [W3C, 2008] W3C Recommendation, “SPARQL Protocol for RDF”, <http://www.w3.org/TR/rdf-sparql-protocol/>, 2008.
- [Wang et al., 2005] Z. Wang, X. Xu, D. Zhan, “A Survey of Business Component Identification Methods and Related Techniques”, *International Journal of Information Technology*, vol 2, pp. 229-238, 2005.
- [WSDL, 2007] Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, <http://www.w3.org/TR/wsdl20/>, 2007.
- [XQuery, 2004] XQuery 1.0 and XPath 2.0 formal semantics. W3C Working Draft, Feb. 2004.
- [Zachos et al., 2007] K. Zachos, N. Maiden, X. Zhu, S. Jones, “Discovering Web Services to Specify More Complete System Requirements”, *In Proceedings of the 19th international conference on Advanced information systems engineering (CAiSE'07)*, Springer-Verlag Berlin, Heidelberg, 2007.
- [Zimmermann et al., 2004] O. Zimmermann, P. Kroghdahl, C. Gee, “Elements of Service-Oriented Analysis and Design”, *IBM developer works*, 2004.
- [Zoukar, 2005] I. Zoukar, MIBE : Méthode d’Ingénierie des Besoins pour l’implantation d’un progiciel de gestion intégré (ERP), Thèse du doctorat, Centre de Recherche en Informatique, Université de Paris 1 Panthéon-Sorbonne, April 2005.

ANNEXE A

1. SERVICES INTENTIONNELS DE LA CARTE *E-CLINIQUE*

1.1. La carte « *e-clinique* »



1.1.1. Les services atomiques

Section de la carte	Service atomique	N°
ab1	<i>S</i> <i>accueillir un patient par planification périodique</i>	<i>S</i> ₁
ab2	<i>S</i> <i>accueillir un patient par planification prévision stratégique</i>	<i>S</i> ₂
ab3	<i>S</i> <i>accueillir un patient par rendez-vous</i>	<i>S</i> ₃
bc1	<i>S</i> <i>diagnostiquer une maladie par examen général</i>	<i>S</i> ₄
cc1	<i>S</i> <i>diagnostiquer une maladie par analyse médicale</i>	<i>S</i> ₅
cc1	<i>S</i> <i>diagnostiquer une maladie par endoscopie</i>	<i>S</i> ₆
cc3	<i>S</i> <i>diagnostiquer une maladie par scanner</i>	<i>S</i> ₇
cc4	<i>S</i> <i>diagnostiquer une maladie par radiologie</i>	<i>S</i> ₈
cd1	<i>S</i> <i>prescrire un traitement par un consultation du dossier médical</i>	<i>S</i> ₉
dd1	<i>S</i> <i>prescrire un traitement par un acte chirurgical</i>	<i>S</i> ₁₀
dd2	<i>S</i> <i>prescrire un traitement par la prescription des médicaments</i>	<i>S</i> ₁₁
de1	<i>S</i> <i>éditer la facture de l'acte médical par calcul automatique de frais</i>	<i>S</i> ₁₂
de2	<i>S</i> <i>éditer la facture de l'acte médical par demande explicite</i>	<i>S</i> ₁₃

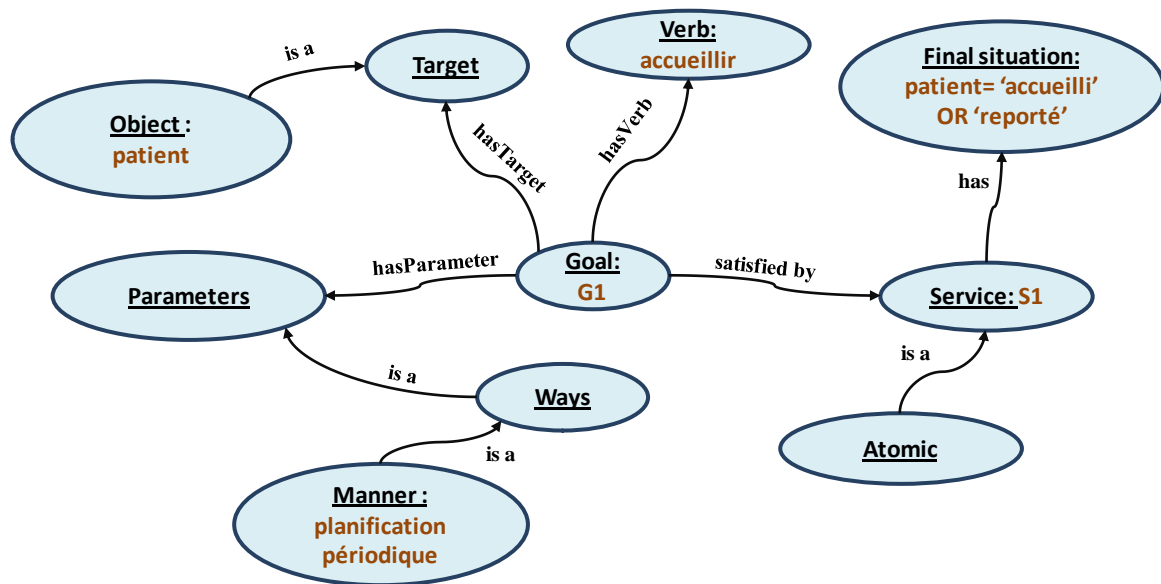
ee1	S <i>éditer la facture de l'acte médical avec prise en charge par un tiers</i>	S_{14}
ee2	S <i>éditer la facture de l'acte médical avec prise en charge par le patient</i>	S_{15}
ef1	S <i>régler la facture par prise en charge complète</i>	S_{16}
ef2	S <i>régler la facture par virement</i>	S_{17}
ef3	S <i>régler la facture par carte bancaire</i>	S_{18}
ef4	S <i>régler la facture par chèque</i>	S_{19}
ef5	S <i>régler la facture en espèce</i>	S_{20}
fg1	S <i>arrêter par clôture du dossier</i>	S_{21}
cg1	S <i>arrêter par demande d'annulation du client</i>	S_{22}

1.1.2. Les services agrégats

Type de relation		Service identifié		N°
Paquet	$P_{de} = \otimes (de_1, de_2)$	Service à choix alternatif	S <i>Préparer la facture de l'acte médical</i>	S_{23}
	$P_{ef} = \otimes (ef_1, ef_2, ef_3, ef_4, ef_5)$		S <i>Régler une facture</i>	S_{24}
Multi-segments	$MS_{ab} = \vee (ab_1, ab_2, ab_3)$	Service à choix multiple	S <i>Accueillir un patient</i>	S_{25}
	$MS_{cc} = \vee (cc_1, cc_2, cc_3, cc_4)$		S <i>Diagnostiquer une maladie d'un patient</i>	S_{26}
	$MS_{dd} = \vee (dd_1, dd_2)$		S <i>Prescrire un traitement d'un patient</i>	S_{27}
	$MS_{ee} = \vee (ee_1, ee_2)$		S <i>Éditer la facture de l'acte médical</i>	S_{28}
Chemin	$C_{e,\{f\},g} = \bullet (MS_{ee}^*, P_{ef}, fg_1)$	Service composite séquentiel	S <i>Régler la facture de l'acte médical</i>	S_{29}
	$C_{d,\{e,f\},g} = \bullet (P_{de}, C_{e,\{f\},g})$		S <i>Facturer un acte médical</i>	S_{30}
	$C_{c,\{d,e,f\},g} = \bullet (cd_1, MS_{dd}^*, C_{d,\{e,f\},g})$		S <i>Réaliser une visite médicale d'un patient</i>	S_{31}
	$C_{a,\{b,c,d,e,f\},g} = \bullet (MS_{ab}, bc_1, MS_{cc}^*, MC_{c,\{d,e,f\},g})$		S <i>Traiter un patient dans la clinique</i>	S_{32}
Multi-chemin	$MC_{c,\{d,e,f\},g} = \cup (cg_1, C_{c,\{d,e,f\},g})$	Service à variation de chemin	S <i>Finaliser une visite médicale d'un patient</i>	S_{33}

1.2. Représentation des services intentionnels et descripteurs

1.2.1. Service atomique S_1



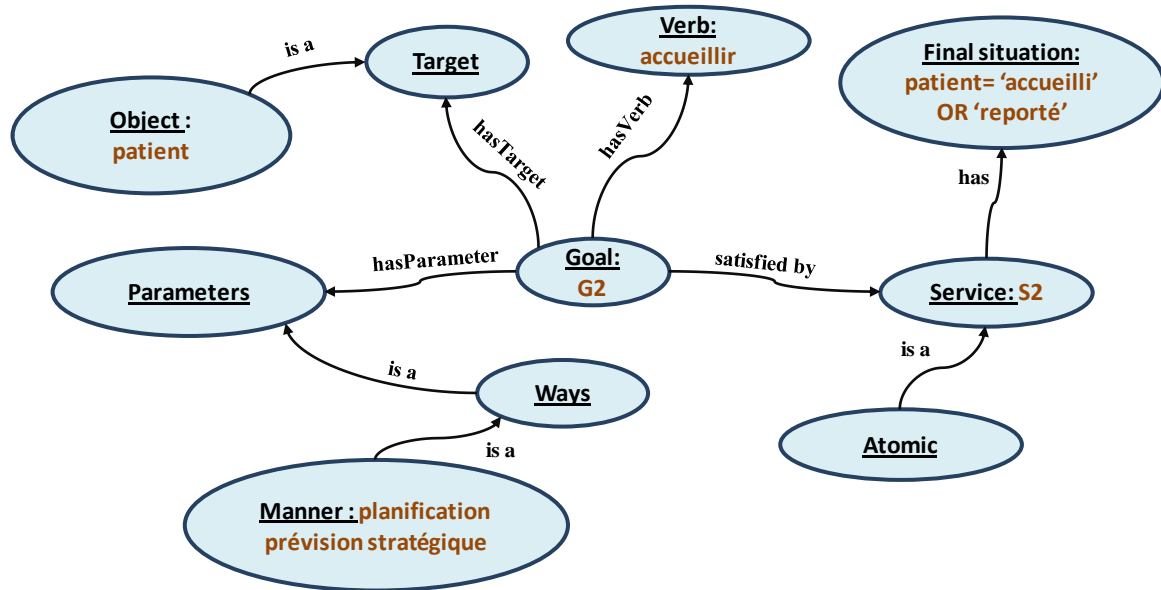
Présentation du service atomique $S_{accueillir\ un\ patient\ par\ planification\ périodique}$ dans iSOnto

```

<service name="accueillir un patient par planification périodique"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#accueillir
    &http://.../HealthOntology;#patient
    &http://.../HealthOntology;#planification périodique
    &http://.../HealthOntology;#patient='accueilli' OR 'reporté'
    &http://.../iServiceOntology;#S1">
</service name>
  
```

L'annotation ajoutée au descripteur du service atomique $S_{accueillir\ un\ patient\ par\ planification\ périodique}$

1.2.2. Service atomique S_2

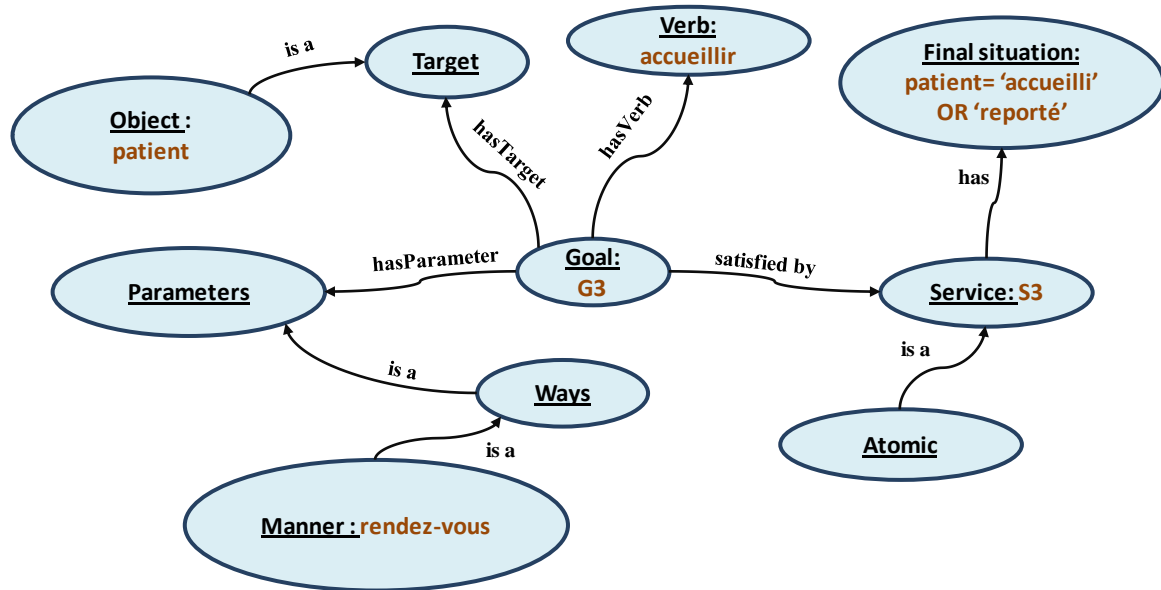


Présentation du service atomique $S_{accueillir\ un\ patient\ par\ planification\ prévision\ stratégique}$ dans iOnto

```
<service name="accueillir un patient par planification prévision stratégique"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#accueillir
    &http://.../HealthOntology;#patient
    &http://.../HealthOntology;#planification prévision stratégique
    &http://.../HealthOntology;#patient='accueilli' OR 'reporté'
    &http://.../iServiceOntology;#S2">
</service name>
```

L'annotation ajoutée au descripteur du service atomique $S_{accueillir\ un\ patient\ par\ planification\ prévision\ stratégique}$

1.2.3. Service atomique S_3



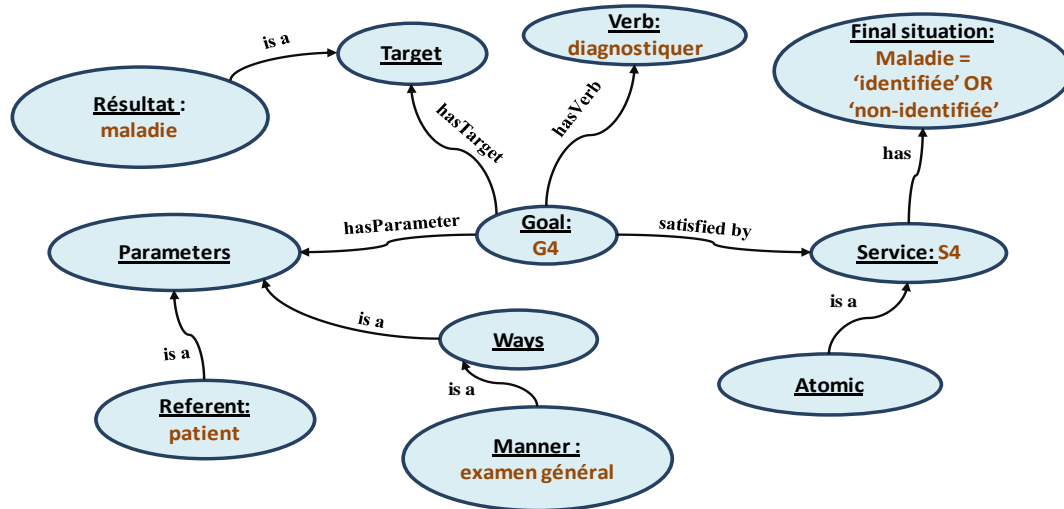
Présentation du service atomique $S_{accueillir\ un\ patient\ par\ rendez-vous}$ dans iSonto

```

<service name="accueillir un patient par rendez-vous"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#accueillir
    &http://.../HealthOntology;#patient
    &http://.../HealthOntology;#rendez-vous
    &http://.../HealthOntology;#patient='accueilli' OR 'reporté'
    &http://.../iServiceOntology;#S3">
</service name>
  
```

L'annotation ajoutée au descripteur du service atomique $S_{accueillir\ un\ patient\ par\ rendez-vous}$

1.2.4. Service atomique S_4

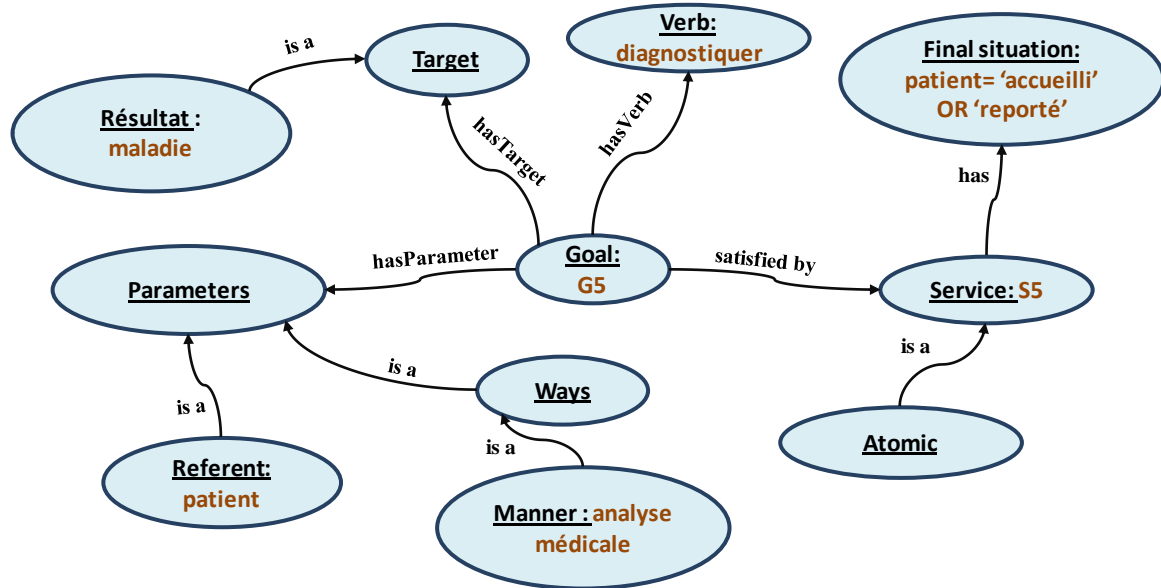


Présentation du service atomique $S_{\text{Diagnostiquer une maladie par examen général}}$ dans iOnto

```
<service name="Diagnostiquer une maladie par examen général"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#Diagnostiquer
    &http://.../HealthOntology;#maladie
    &http://.../HealthOntology;#patient
    &http://.../HealthOntology;#examen_medical
    &http://.../HealthOntology;#patient='accueilli' OR 'reporté'
    &http://.../iServiceOntology;#S4">
</service name>
```

L'annotation ajoutée au descripteur du service à variation $S_{\text{Diagnostiquer une maladie par examen général}}$

1.2.5. Service atomique S_5



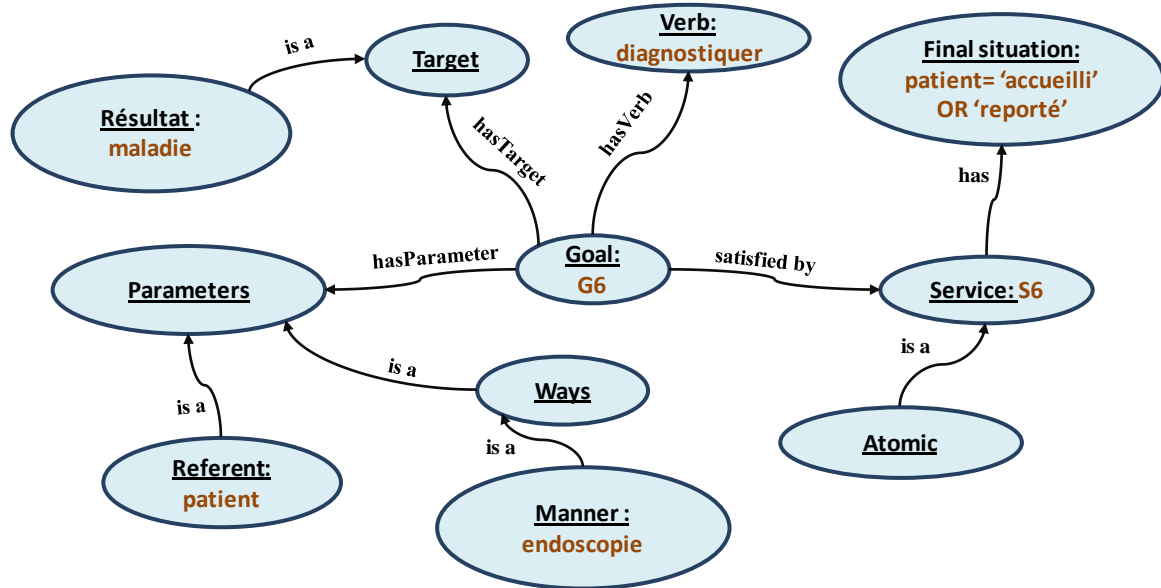
Présentation du service atomique $S_{\text{Diagnostiquer une maladie par analyse médicale}}$ dans iOnto

```

<service name="Diagnostiquer une maladie par analyse médicale"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#Diagnostiquer
    &http://.../HealthOntology;#maladie
    &http://.../HealthOntology;#patient
    &http://.../HealthOntology;#analyse médicale
    &http://.../HealthOntology;#patient='accueilli' OR 'reporté'
    &http://.../iServiceOntology;#S5">
</service name>
  
```

L'annotation ajoutée au descripteur du service à variation $S_{\text{Diagnostiquer une maladie par analyse médicale}}$

1.2.6. Service atomique S_6



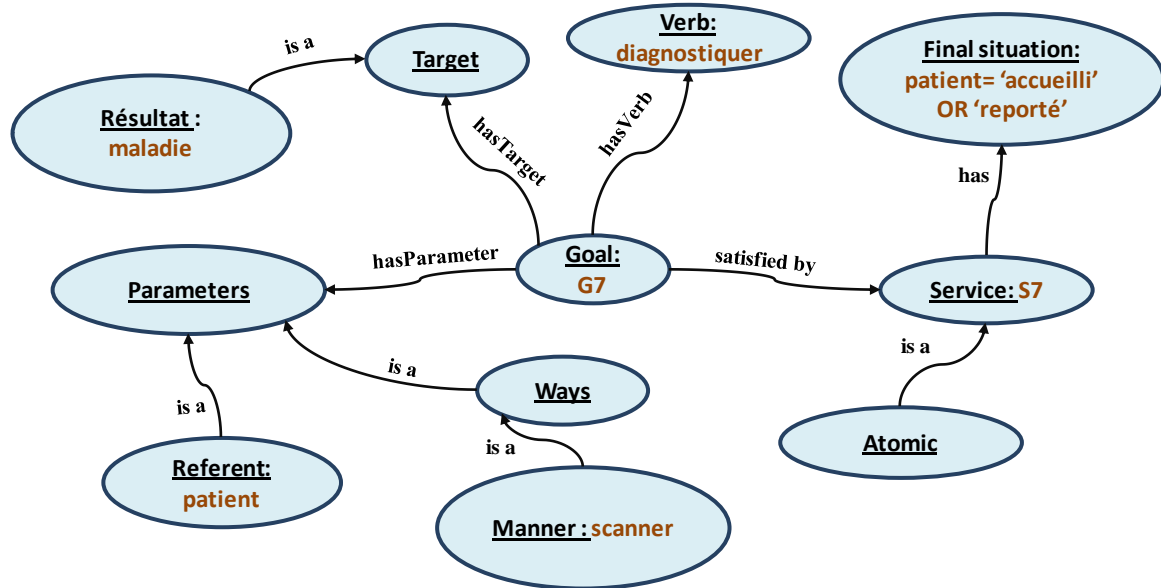
Présentation du service atomique $S_{\text{Diagnostiquer une maladie par endoscopie}}$ dans iOnto

```

<service name="Diagnostiquer une maladie par endoscopie"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#Diagnostiquer
    &http://.../HealthOntology;#maladie
    &http://.../HealthOntology;#patient
    &http://.../HealthOntology;#endoscopie
    &http://.../HealthOntology;#patient='accueilli' OR 'reporté'
    &http://.../iServiceOntology;#S6">
</service name>
  
```

L'annotation ajoutée au descripteur du service à variation $S_{\text{Diagnostiquer une maladie par endoscopie}}$

1.2.7. Service atomique S_7

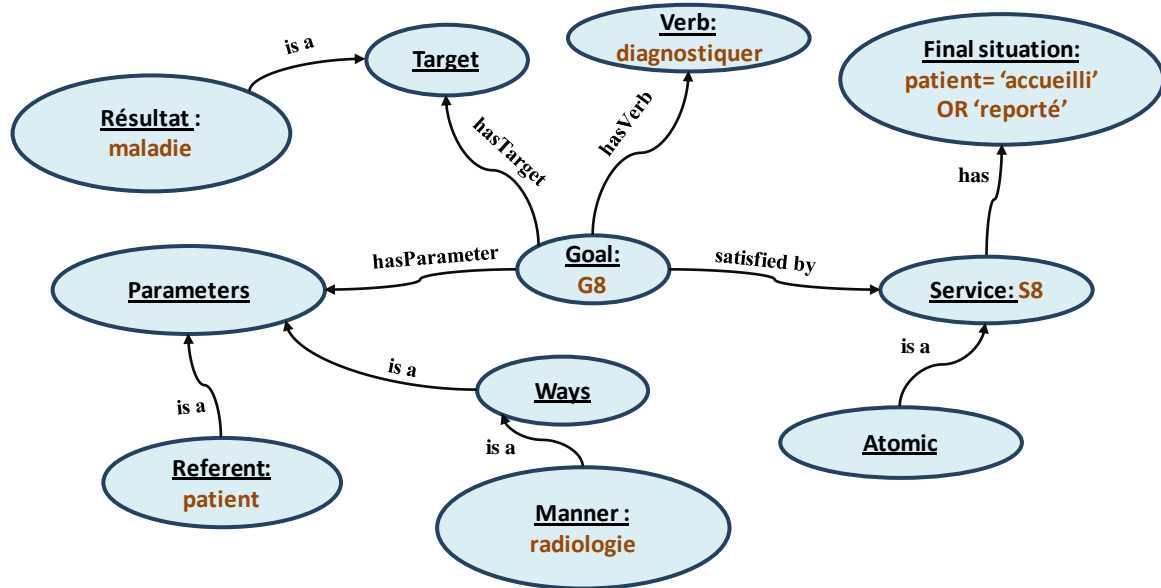


Présentation du service atomique $S_{\text{Diagnostiquer une maladie par scanner}}$ dans iSOnto

```
<service name="Diagnostiquer une maladie par scanner"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://... /VerbOntology;#Diagnostiquer
    &http://... /HealthOntology;#maladie
    &http://... /HealthOntology;#patient
    &http://... /HealthOntology;#scanner
    &http://.../HealthOntology;#patient='accueilli' OR 'reporté'
    &http://... / iServiceOntology;#S7">
</service name>
```

L'annotation ajoutée au descripteur du service à variation $S_{\text{Diagnostiquer une maladie par scanner}}$

1.2.8. Service atomique S_8



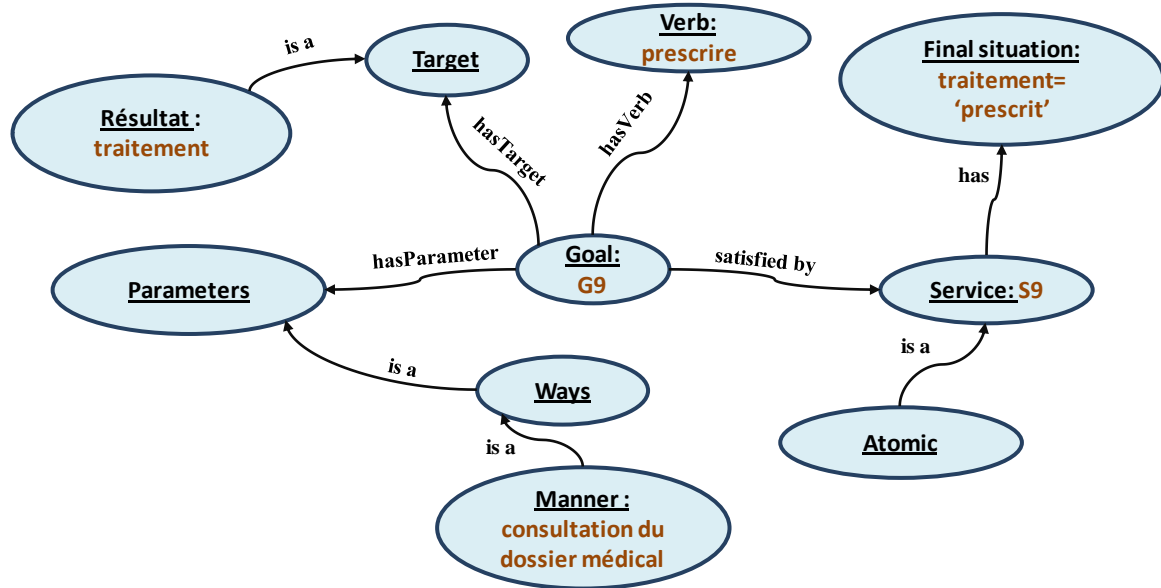
Présentation du service atomique $S_{\text{Diagnostiquer une maladie par radiologie}}$ dans iOnto

```

<service name="Diagnostiquer une maladie par radiologie"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#Diagnostiquer
    &http://.../HealthOntology;#maladie
    &http://.../HealthOntology;#patient
    &http://.../HealthOntology;#radiologie
    &http://.../HealthOntology;#patient='accueilli' OR 'reporté'
    &http://.../iServiceOntology;#S8">
</service name>
  
```

L'annotation ajoutée au descripteur du service à variation $S_{\text{Diagnostiquer une maladie par radiologie}}$

1.2.9. Service atomique S_9



Présentation du service atomique $S_{prescrire\ un\ traitement\ par\ consultation\ du\ dossier\ médical}$ dans iSonto

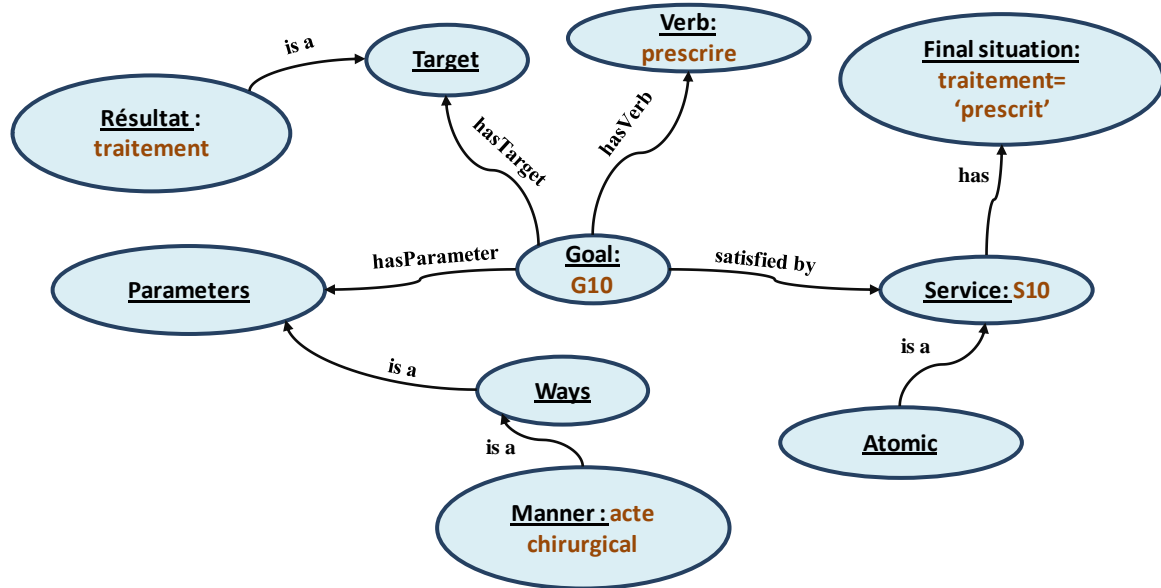
```

<service name="prescrire un traitement par consultation du dossier médical"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#prescrire
    &http://.../HealthOntology;#traitement
    &http://.../HealthOntology;#consultation du dossier médical
    &http://.../HealthOntology;#traitement='prescrit'
    &http://.../iServiceOntology;#S9">
</service name>

```

L'annotation ajoutée au descripteur du service à variation $S_{prescrire\ un\ traitement\ par\ consultation\ du\ dossier\ médical}$

1.2.10. Service atomique S_{10}



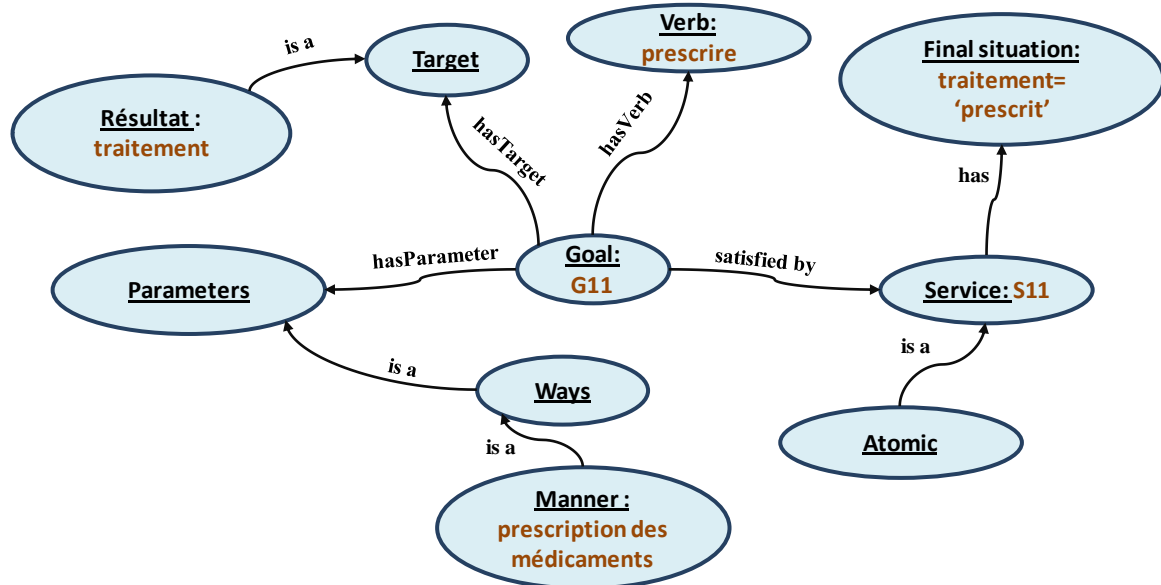
Présentation du service atomique $S_{prescrire\ un\ traitement\ par\ acte\ médical}$ dans iSOnto

```

<service name="prescrire un traitement par acte médical"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#prescrire
    &http://.../HealthOntology;#traitement
    &http://.../HealthOntology;#acte médical
    &http://.../HealthOntology;#traitement='prescrit'
    &http://.../iServiceOntology;#S10">
</service name>
  
```

L'annotation ajoutée au descripteur du service à variation $S_{prescrire\ un\ traitement\ par\ acte\ médical}$

1.2.11. Service atomique S_{11}

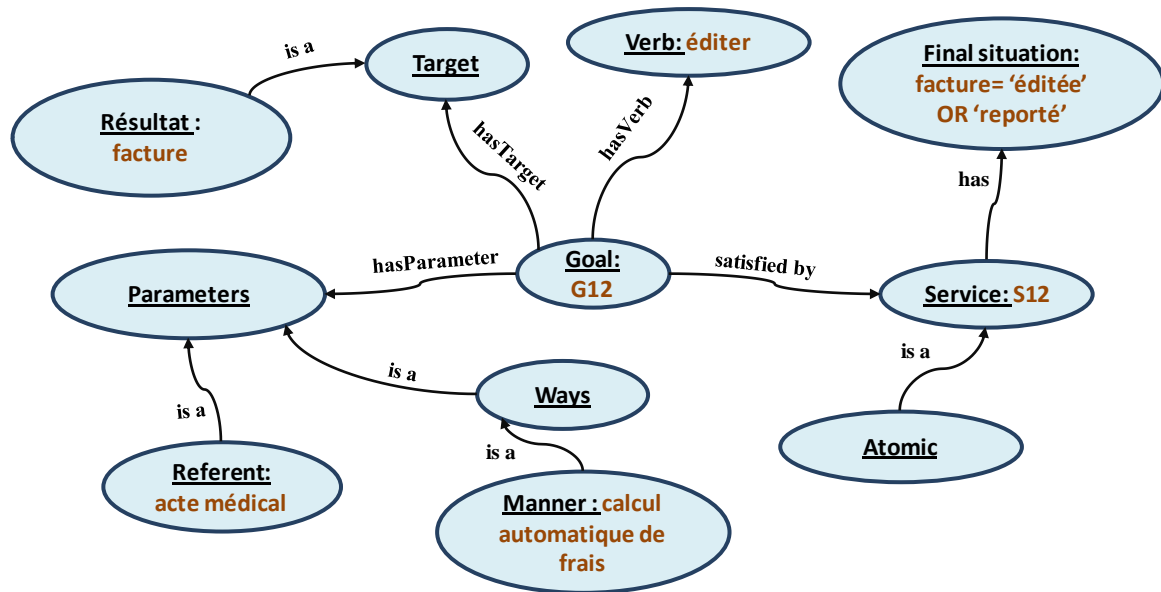


Présentation du service atomique $S_{prescrire\ un\ traitement\ par\ prescription\ des\ médicaments}$ dans iServiceOntology

```
<service name="prescrire un traitement par prescription des médicaments"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#prescrire
    &http://.../HealthOntology;#traitement
    &http://.../HealthOntology;#prescription\ des\ médicaments
    &http://.../HealthOntology;#traitement='prescrit'
    &http://.../iServiceOntology;#S11">
</service name>
```

L'annotation ajoutée au descripteur du service à variation $S_{prescrire\ un\ traitement\ par\ prescription\ des\ médicaments}$

1.2.12. Service atomique S_{12}

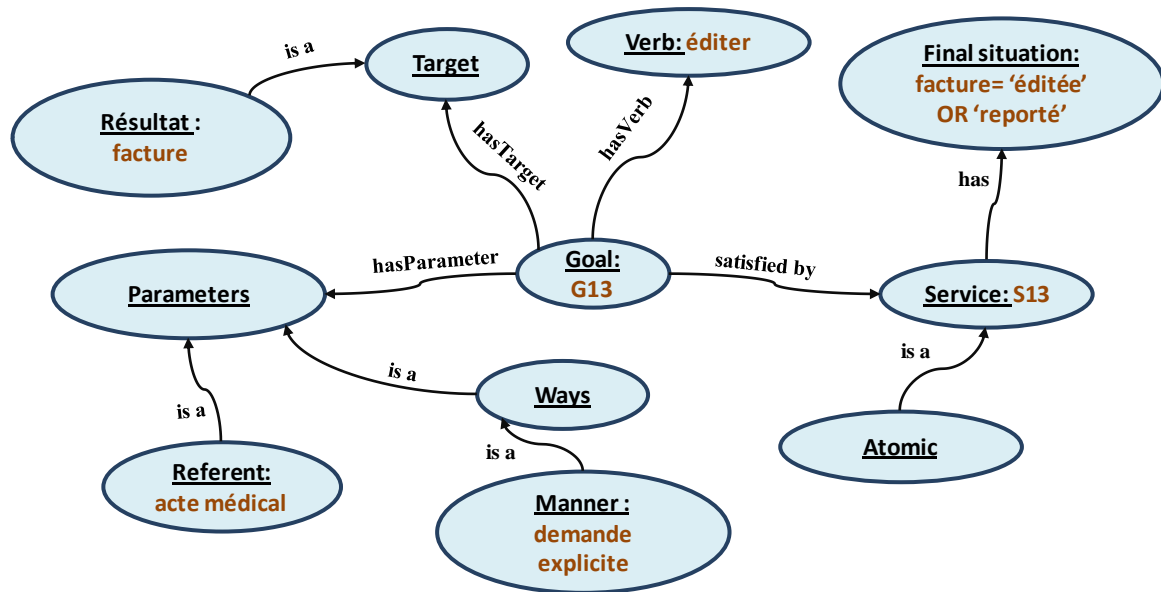


Présentation du service atomique S éditer la facture de l'acte médical par calcul automatique de frais dans iServiceOntology

```
<service name="éditer la facture de l'acte médical par calcul automatique de frais"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#éditer
    &http://.../HealthOntology;#facture
    &http://.../HealthOntology;#acte médical
    &http://.../HealthOntology;#calcul automatique de frais
    &http://.../HealthOntology;#facture='éditée' OR 'reportée'
    &http://.../iServiceOntology;#S12">
</service name>
```

L'annotation ajoutée au descripteur du service à variation S éditer la facture de l'acte médical par calcul automatique de frais

1.2.13. Service atomique S_{12}

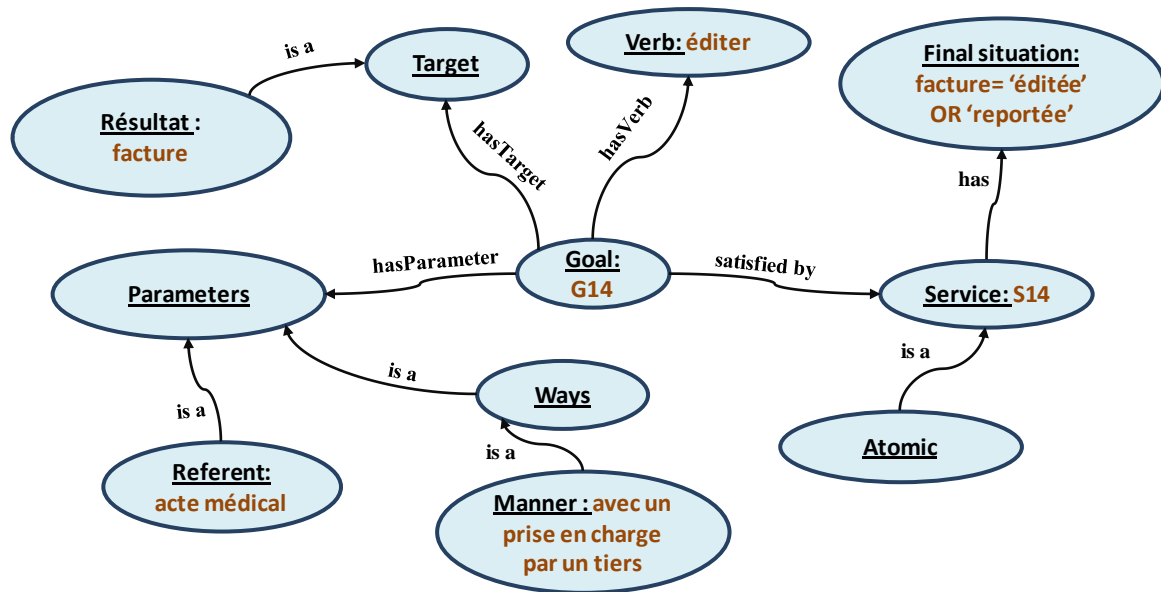


Présentation du service atomique S éditer la facture de l'acte médical par demande explicite dans iServiceOntology

```
<service name="éditer la facture de l'acte médical par demande explicite"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#éditer
    &http://.../HealthOntology;#facture
    &http://.../HealthOntology;#acte médical
    &http://.../HealthOntology;#demande explicite
    &http://.../HealthOntology;#facture='éditée' OR 'reportée'
    &http://.../iServiceOntology;#S13">
</service name>
```

L'annotation ajoutée au descripteur du service à variation S éditer la facture de l'acte médical par demande explicite

1.2.14. Service atomique S_{14}



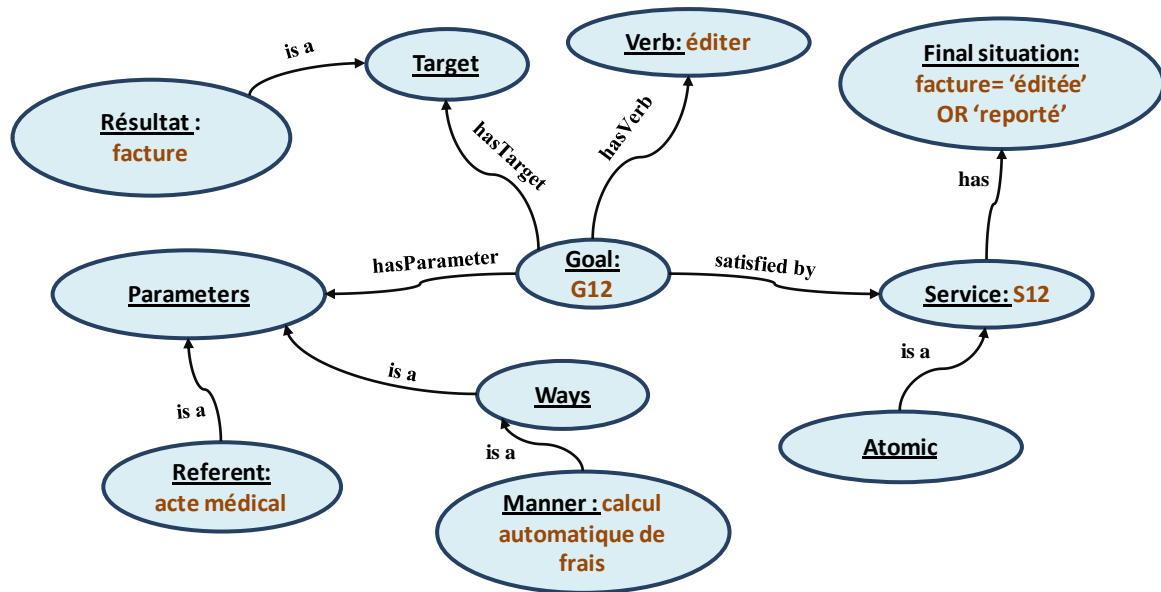
Présentation du service atomique S éditer la facture de l'acte médical avec prise en charge par un tiers dans iSOnto

```

<service name="éditer la facture de l'acte médical avec prise en charge par un tiers"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#éditer
    &http://.../HealthOntology;#facture
    &http://.../HealthOntology;#acte médical
    &http://.../HealthOntology;#avec prise en charge par un tiers
    &http://.../HealthOntology;#facture='éditée' OR 'reportée'
    &http://.../iServiceOntology;#S14">
  </service name>
  
```

L'annotation ajoutée au descripteur du service à variation S éditer la facture de l'acte médical avec prise en charge par un tiers

1.2.15. Service atomique S_{15}



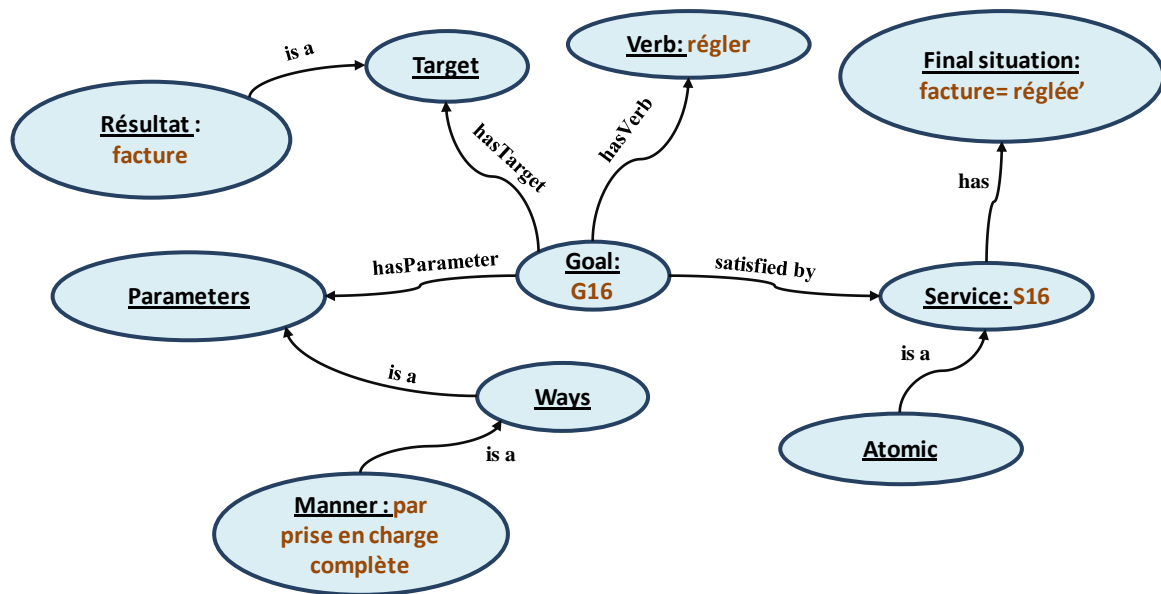
Présentation du service atomique S éditer la facture de l'acte médical avec prise en charge par le patient dans iServiceOntology

```

<service name="éditer la facture de l'acte médical avec prise en charge par le patient"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://.../VerbOntology;#éditer
    &http://.../HealthOntology;#facture
    &http://.../HealthOntology;#acte médical
    &http://.../HealthOntology;# avec prise en charge par le patient
    &http://.../HealthOntology;#facture='éditée' OR 'reportée'
    &http://.../iServiceOntology;#S15">
</service name>
  
```

L'annotation ajoutée au descripteur du service à variation S éditer la facture de l'acte médical avec prise en charge par le patient

1.2.16. Service atomique S_{16}



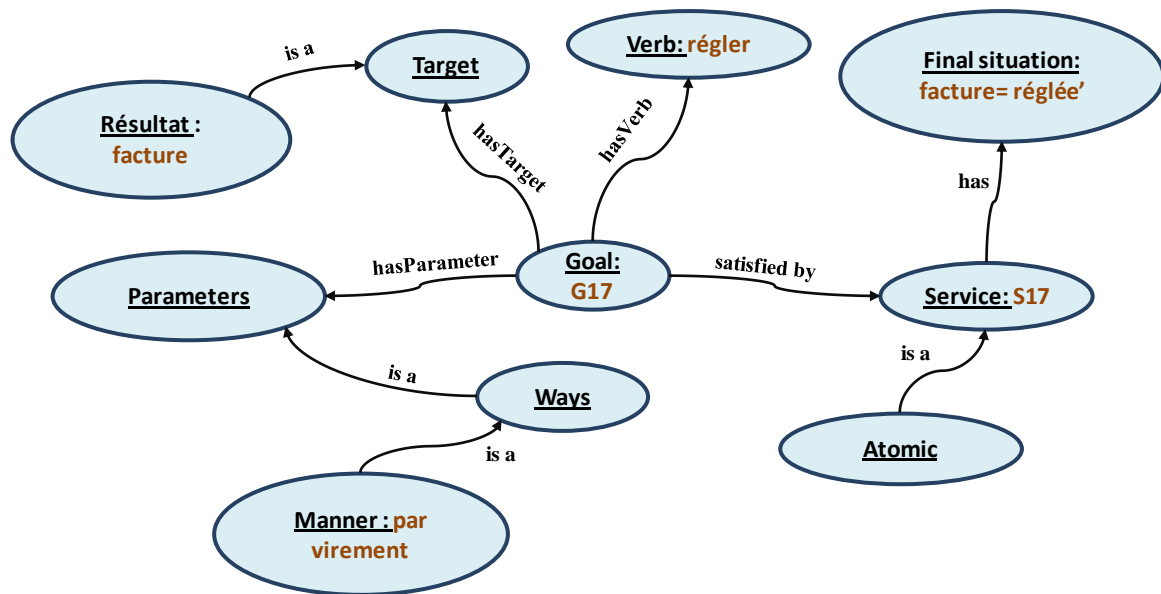
Présentation du service atomique S régler la facture par prise en charge complète dans iOnto

```

<service name="régler la facture par prise en charge complète"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://... /VerbOntology;#régler
    &http://... /HealthOntology;#facture
    &http://... /HealthOntology;#prise en charge complète
    &http://.../HealthOntology;#facture='réglée'
    &http://... / iServiceOntology;#S16">
</service name>
  
```

L'annotation ajoutée au descripteur du service à variation S régler la facture par prise en charge complète

1.2.17. Service atomique S_{17}



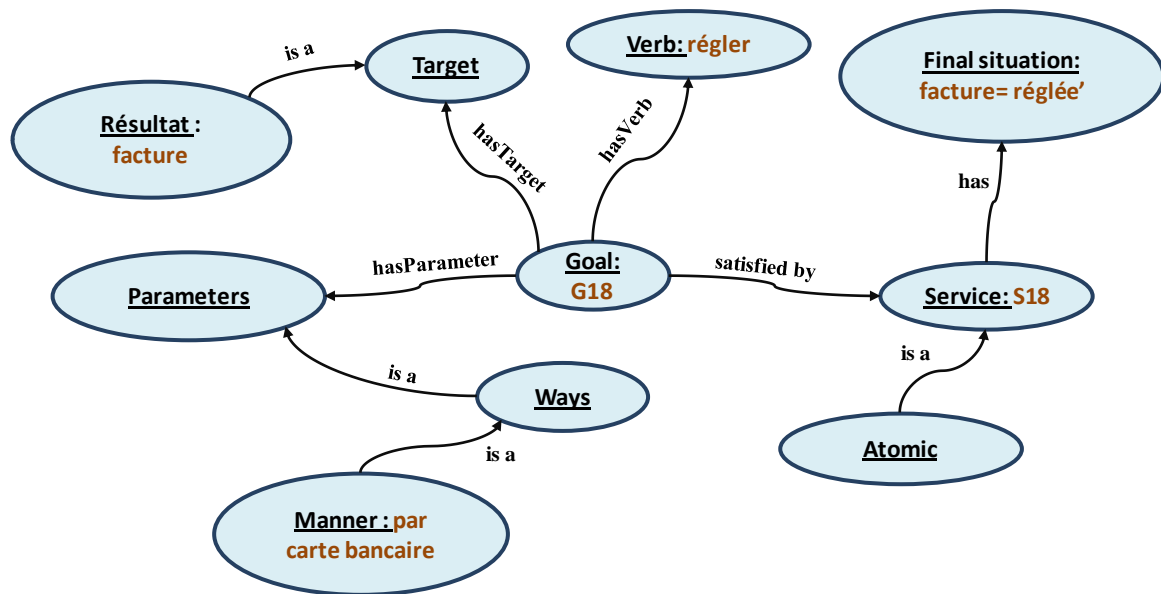
Présentation du service atomique $S_{\text{régler la facture par virement}}$ dans iOnto

```

<service name="régler la facture par virement"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://... /VerbOntology;#régler
    &http://... /HealthOntology;#facture
    &http://... /HealthOntology;#virement
    &http://... /HealthOntology;#facture='réglée'
    &http://... / iServiceOntology;#S16">
</service name>
  
```

L'annotation ajoutée au descripteur du service à variation $S_{\text{régler la facture par virement}}$

1.2.18. Service atomique S_{18}



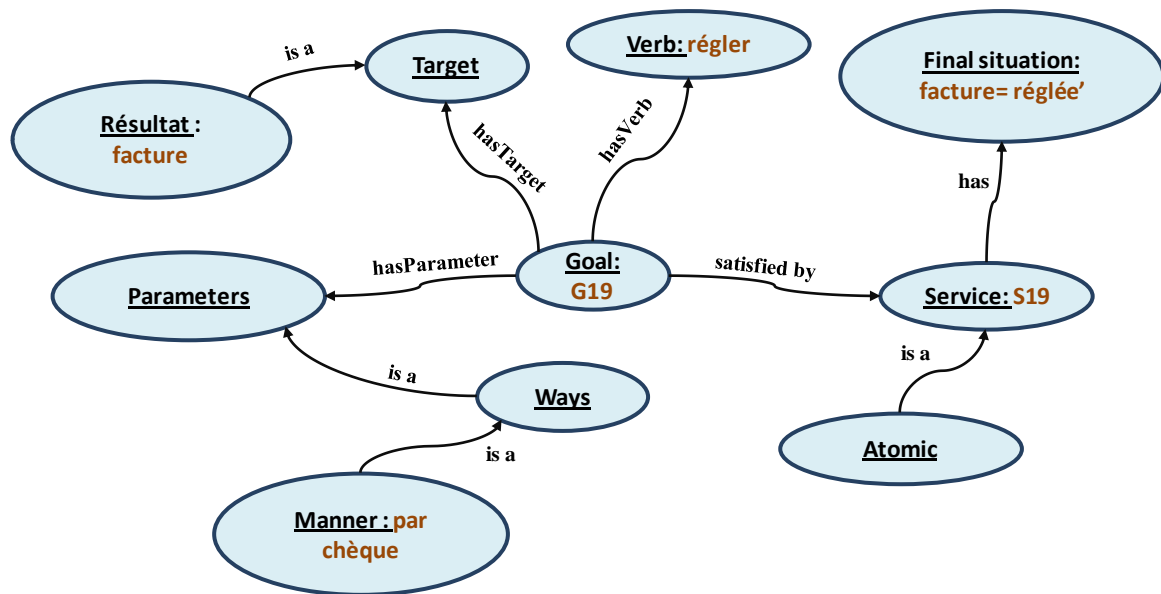
Présentation du service atomique $S_{\text{régler la facture par carte bancaire}}$ dans iOnto

```

<service name="régler la facture par carte bancaire"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://... /VerbOntology;#régler
    &http://... /HealthOntology;#facture
    &http://... /HealthOntology;#carte bancaire
    &http://... /HealthOntology;#facture='réglée'
    &http://... / iServiceOntology;#S18">
</service name>
  
```

L'annotation ajoutée au descripteur du service à variation $S_{\text{régler la facture par prise en charge complète}}$

1.2.19. Service atomique S_{19}



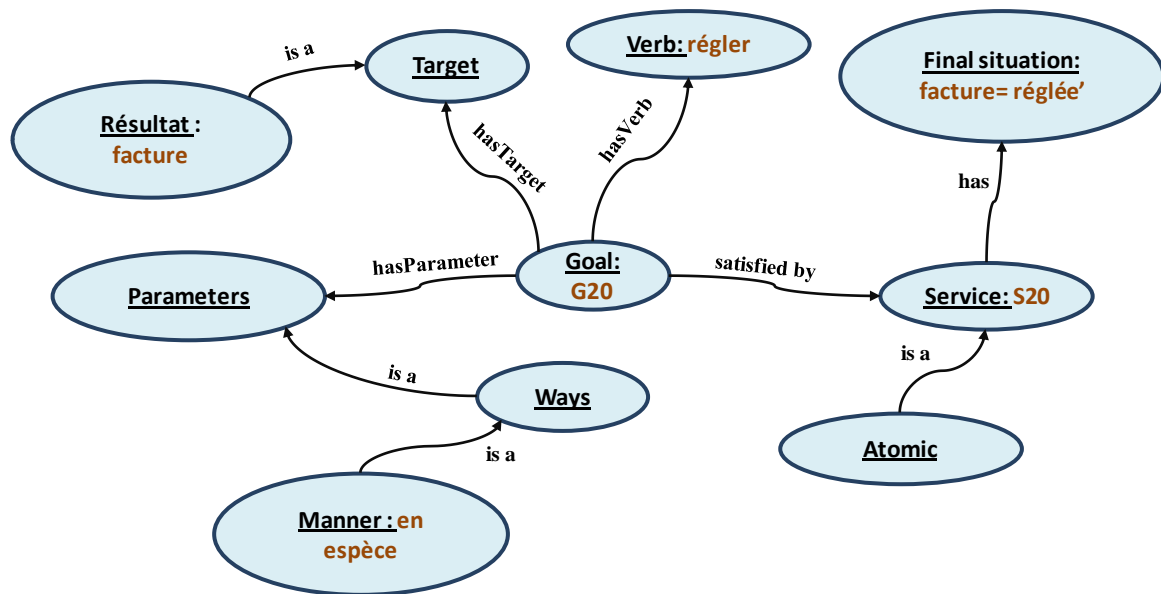
Présentation du service atomique $S_{\text{régler la facture par chèque}}$ dans iOnto

```

<service name="régler la facture par chèque"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://... /VerbOntology;#régler
    &http://... /HealthOntology;#facture
    &http://... /HealthOntology;#chèque
    &http://.../HealthOntology;#facture='réglée'
    &http://... / iServiceOntology;#S19">
</service name>
  
```

L'annotation ajoutée au descripteur du service à variation $S_{\text{régler la facture par chèque}}$

1.2.20. Service atomique S_{20}



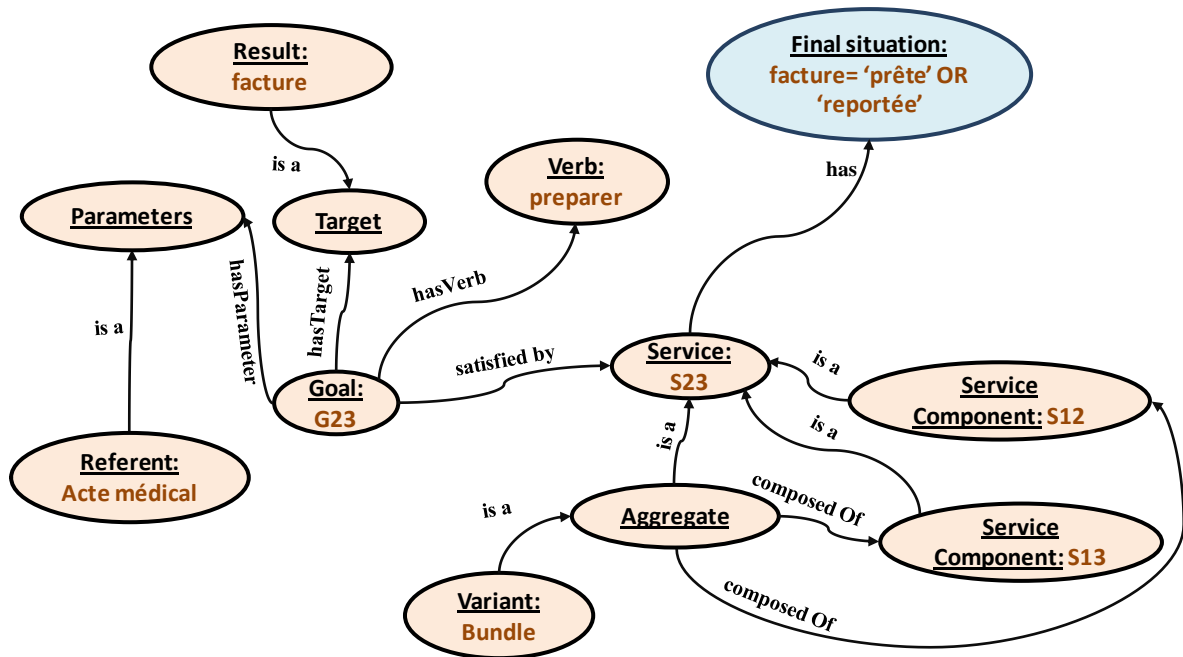
Présentation du service atomique S régler la facture en espèce dans iOnto

```

<service name="régler la facture en espèce"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#manner
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Atomic"
  modelReference="&http://... /VerbOntology;#régler
    &http://... /HealthOntology;#facture
    &http://... /HealthOntology;#espèce
    &http://.../HealthOntology;#facture='réglée'
    &http://... / iServiceOntology;#S20">
</service name>
  
```

L'annotation ajoutée au descripteur du service à variation S régler la facture en espèce

1.2.21. Service agrégat S_{23}

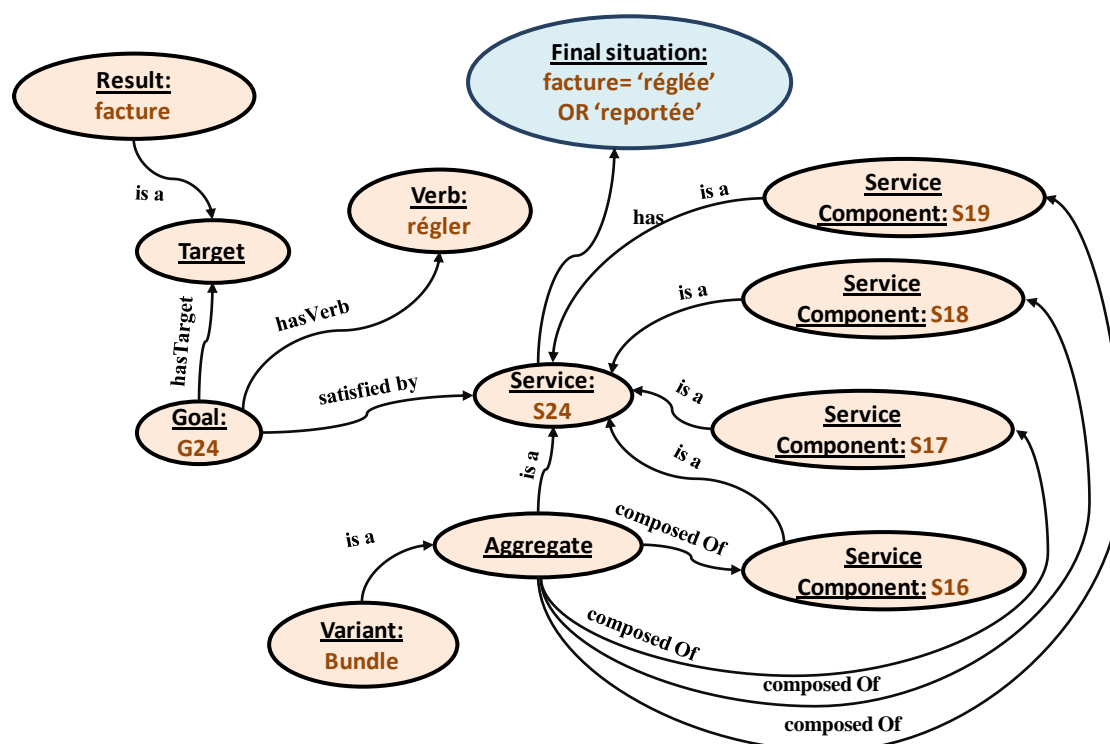


Présentation du service à variation $S_{préparer\ la\ facture\ de\ l'acte\ médical}$ dans iSonto

```
<service name="préparer la facture de l'acte médical"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Bundle"
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
  modelReference="&http://.../VerbOntology;#préparer
    &http://.../HealthOntology;#facture
    &http://.../HealthOntology;#acte médical
    &http://.../HealthOntology;#facture='prête' OR 'reportée'
    &http://.../iServiceOntology;#S23
    &http://.../iServiceOntology;#S12
    &http://.../iServiceOntology;#S13">
</service name>
```

Le descripteur du service à variation $S_{préparer\ la\ facture\ de\ l'acte\ médical}$

1.2.22. Service agrégat S_{24}



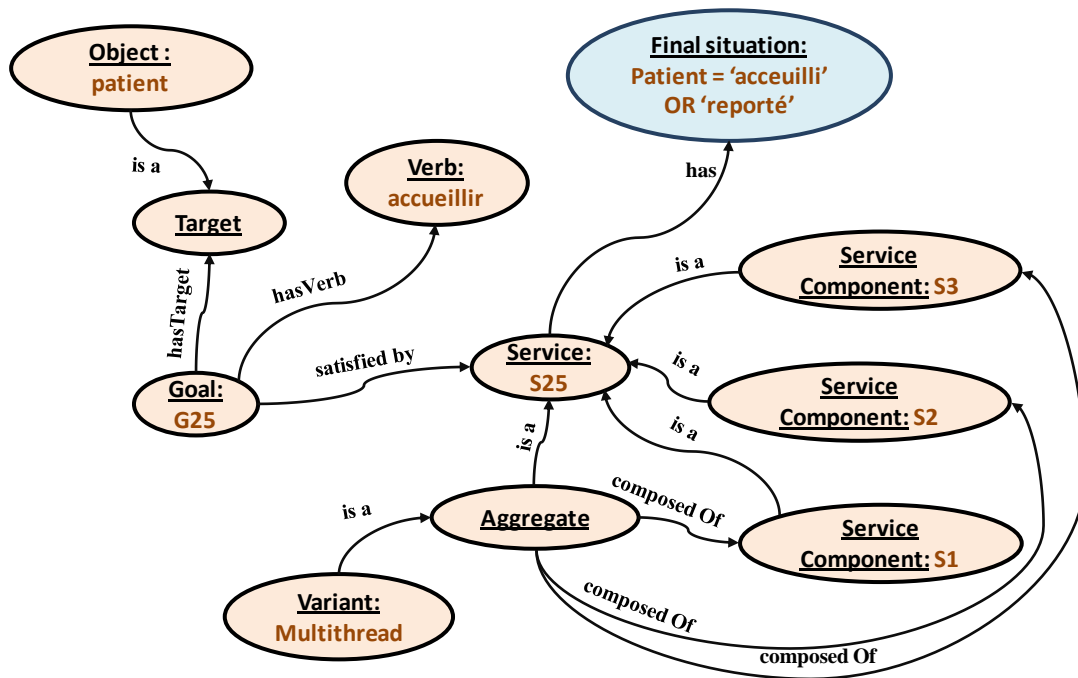
Présentation du service à variation $S_{\text{régler une facture}}$ dans iOnto

```

<service name="régler une facture"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Bundle"
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent"
  modelReference="&http://.../VerbOntology;#régler
    &http://.../HealthOntology;#facture
    &http://.../HealthOntology;#facture='réglée' OR 'reporté'
    &http://.../iServiceOntology;#S24
    &http://.../iServiceOntology;#S16
    &http://.../iServiceOntology;#S17
    &http://.../iServiceOntology;#S18
    &http://.../iServiceOntology;#S19">
</service name>
  
```

Le descripteur du service à variation $S_{\text{régler une facture}}$

1.2.23. Service agrégat S_{25}

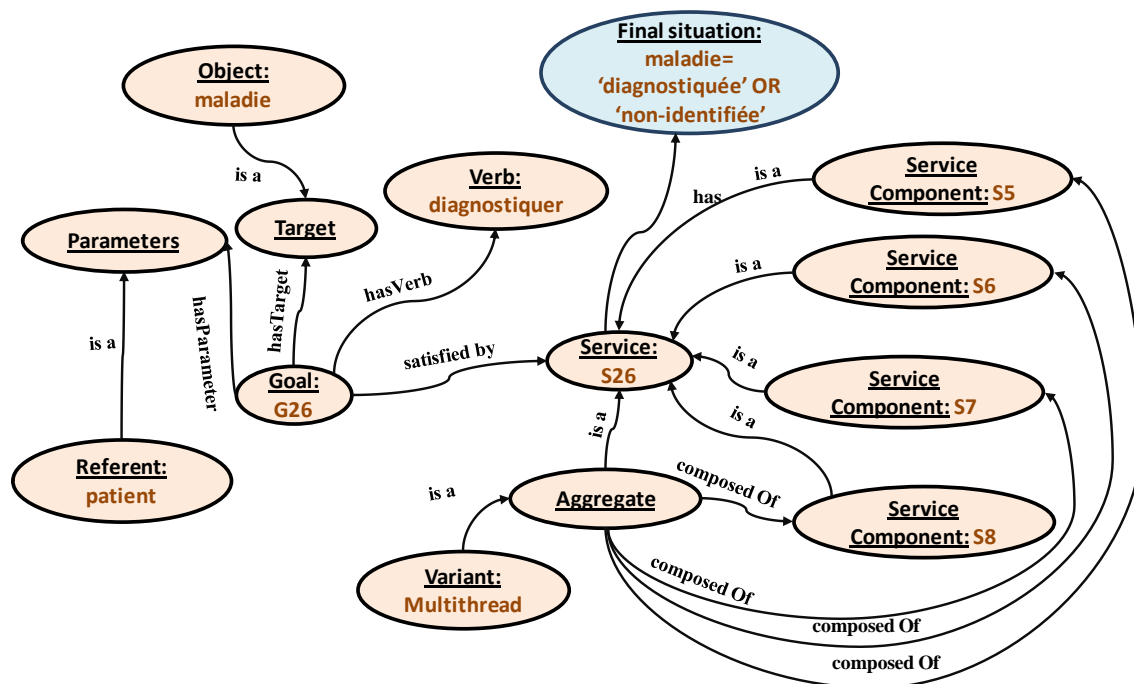


Présentation du service à variation $S_{accueillir\ un\ patient}$ dans iOnto

```
<service name="accueillir un patient"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Multithread"
  modelReference="&http://.../VerbOntology;#accueillir
    &http://.../HealthOntology;#patient
    &http://.../HealthOntology;#patient='accueilli' OR 'reporté'
    &http://.../iServiceOntology;#S25
    &http://.../iServiceOntology;#S1
    &http://.../iServiceOntology;#S2
    &http://.../iServiceOntology;#S3">
</service name>
```

Le descripteur du service à variation $S_{accueillir\ un\ patient}$

1.2.24. Service agrégat S_{26}

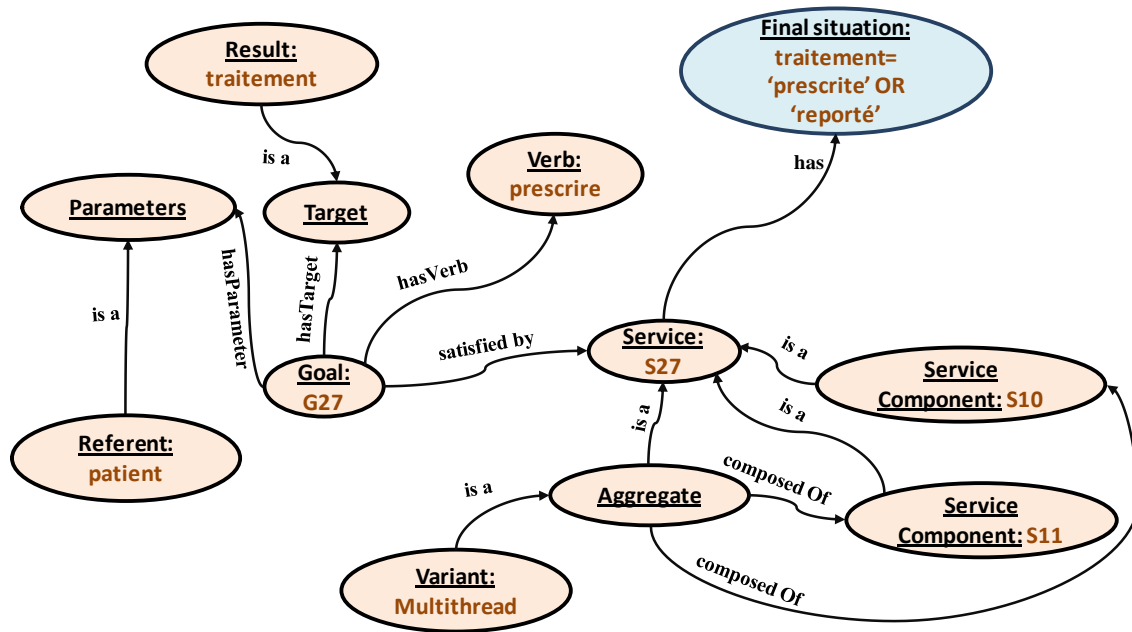


Présentation du service à variation S diagnostiquer une maladie d'un patient dans iSOnto

```
<service name="diagnostiquer une maladie d'un patient"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#Multithread"
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
  modelReference="&http://.../VerbOntology;#diagnostiquer
    &http://.../HealthOntology;#maladie
    &http://.../HealthOntology;#patient
    &http://.../HealthOntology;#maladie='diagnostiquée'
    &http://.../iServiceOntology;#S26
    &http://.../iServiceOntology;#S5
    &http://.../iServiceOntology;#S6
    &http://.../iServiceOntology;#S7
    &http://.../iServiceOntology;#S8">
</service name>
```

Le descripteur du service à variation S diagnostiquer une maladie d'un patient

1.2.25. Service agrégat S_{27}



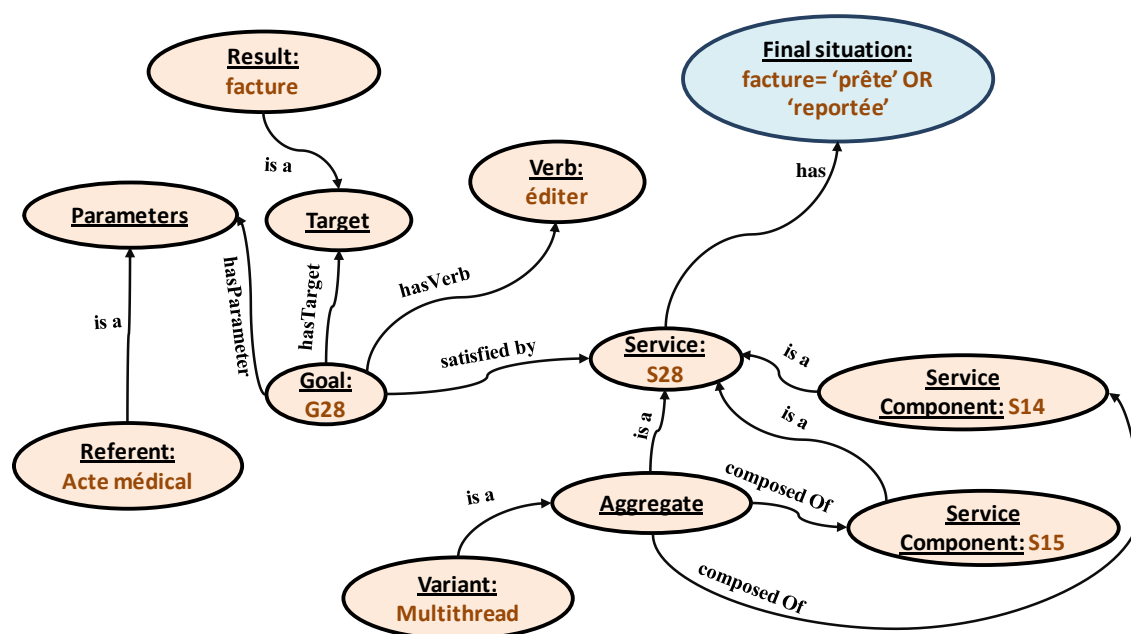
Présentation du service à variation S prescrire un traitement d'un patient dans iSonto

```

<service name="prescrire un traitement d'un patient"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#Multithread"
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
  modelReference="&http://... /VerbOntology;#prescrire
    &http://... /HealthOntology;#traitement
    &http://... /HealthOntology;#patient
    &http://.../HealthOntology;#traitement='prescrire'
    &http://... /iServiceOntology;#S27
    &http://... /iServiceOntology;#S10
    &http://... /iServiceOntology;#S11">
</service name>
  
```

Le descripteur du service à variation S prescrire un traitement d'un patient

1.2.26. Service agrégat S_{28}



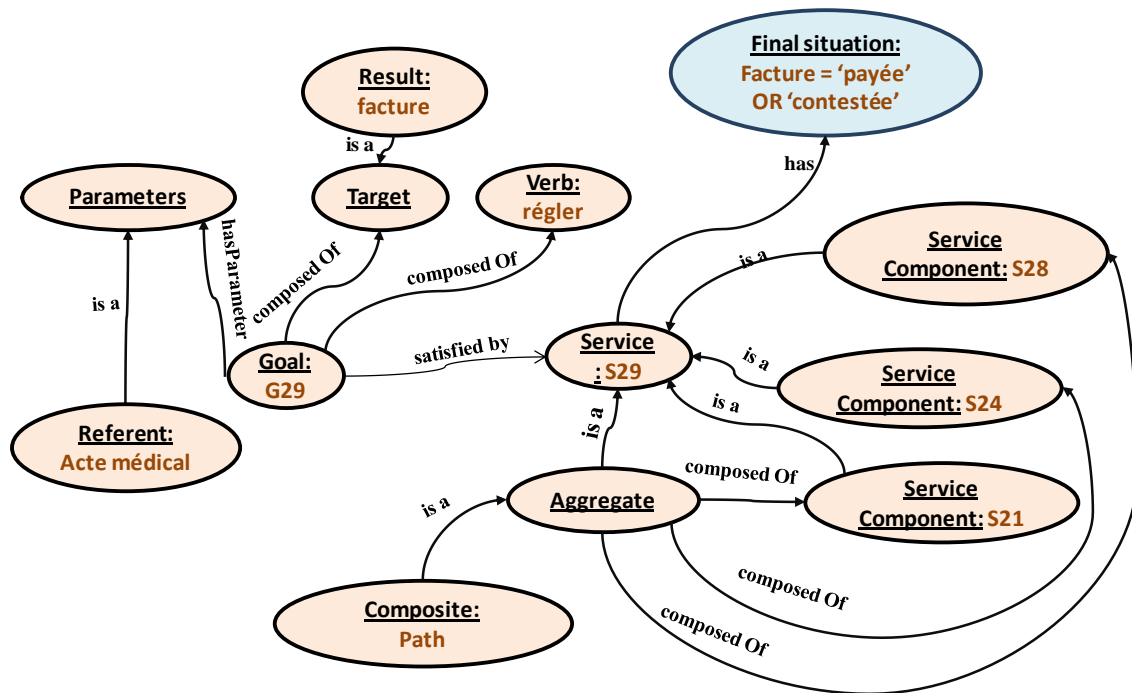
Présentation du service à variation S éditer la facture de l'acte médical dans iSonto

```

<service name="éditer la facture de l'acte médical"
  iServiceConcept="&http://.../iServiceOntology;#Verb
    &http://.../iServiceOntology;#product
    &http://.../iServiceOntology;#final_Stat
    &http://.../iServiceOntology;#Referent
    &http://.../iServiceOntology;#Multithread"
    &http://.../iServiceOntology;#ServiceComponent
    &http://.../iServiceOntology;#ServiceComponent
  modelReference="&http://... /VerbOntology;#éditer
    &http://... /HealthOntology;#facture
    &http://... /HealthOntology;#acte médical
    &http://.../HealthOntology;#facture='prête'
    &http://... /iServiceOntology;#S28
    &http://... /iServiceOntology;#S14
    &http://... /iServiceOntology;#S15">
</service name>
  
```

Le descripteur du service à variation S éditer la facture de l'acte médical

1.2.27. Service agrégat S₂₉



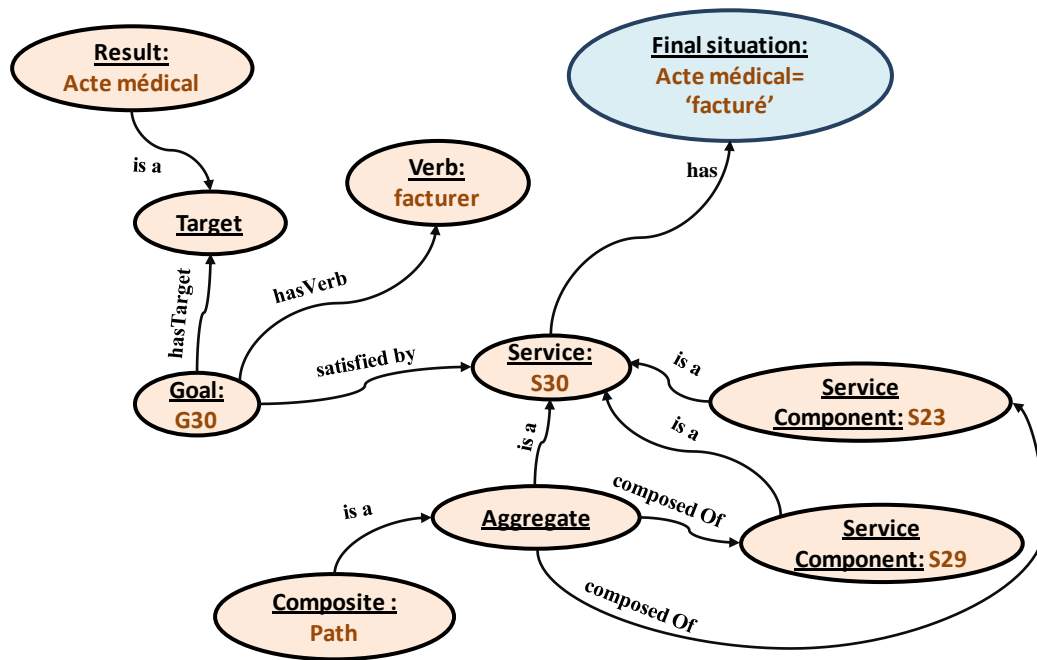
Présentation du service composite S Régler la facture de l'acte médical dans iOnto

```

<service name="Régler la facture de l'acte médical"
  iServiceConcept="http://.../iServiceOntology#Verb
    http://.../iServiceOntology#Product
    http://.../iServiceOntology#Referent
    http://.../iServiceOntology#final_stat
    http://.../iServiceOntology#Path
    http://.../iServiceOntology#ServiceComponent
    http://.../iServiceOntology#ServiceComponent
    http://.../iServiceOntology#ServiceComponent
  modelReference="http://.../VerbOntology#régler
    http://.../HealthOntology#facture
    http://.../HealthOntology#acte médical
    http://.../HealthOntology#facture='réglée' OR 'contestée'
    http://.../iServiceOntology#S29
    http://.../ iServiceOntology#S28
    http://.../ iServiceOntology#S24
    http://.../ iServiceOntology#S21">
</service name>
  
```

Le descripteur du service composite S Régler la facture de l'acte médical

1.2.28. Service agrégat S_{30}

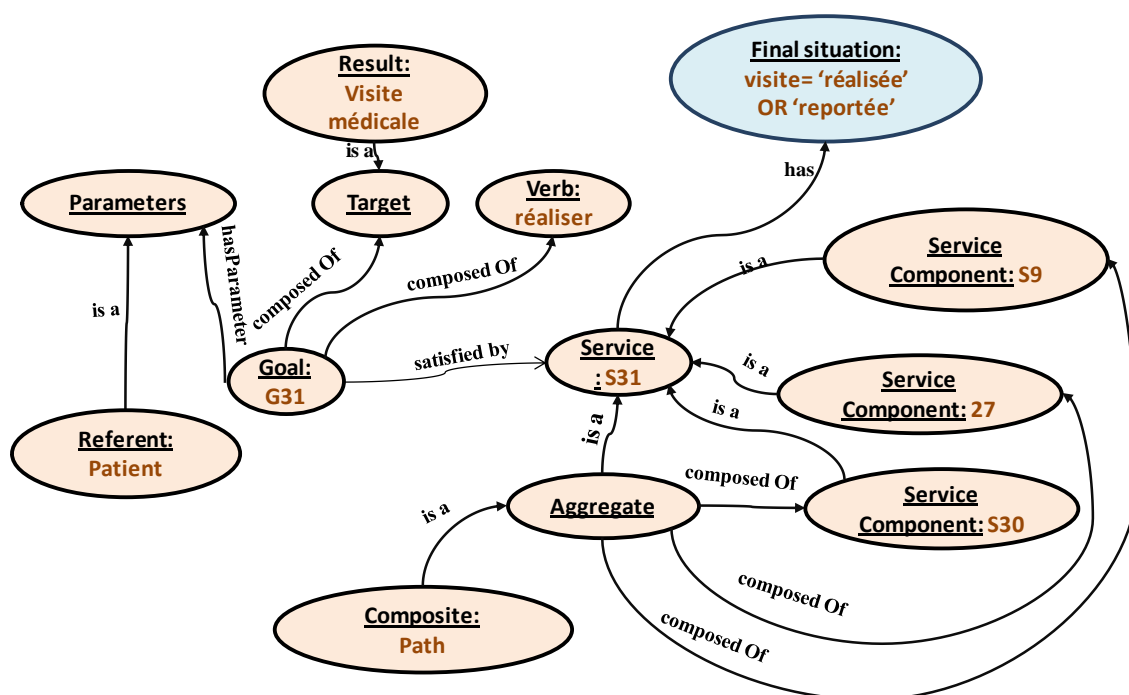


Présentation du service composite $S_{facturer\ l'acte\ médical}$ dans iSOnto

```
<service name="facturer l'acte médical"
  iServiceConcept="http://.../iServiceOntology#Verb
    http://.../iServiceOntology;#Product
    http://.../iServiceOntology;#final_stat
    http://.../iServiceOntology;#Path
    http://.../iServiceOntology;#ServiceComponent
    http://.../iServiceOntology;#ServiceComponent
  modelReference="http://.../VerbOntology;#facturer
    http://.../HealthOntology;#acte médical
    http://.../HealthOntology;#acte médical='facturé'
    http://.../iServiceOntology;#S30
    http://.../iServiceOntology;#S23
    http://.../iServiceOntology;#S29">
</service name>
```

Le descripteur du service composite $S_{facturer\ l'acte\ médical}$

1.2.29. Service agrégat S_{31}

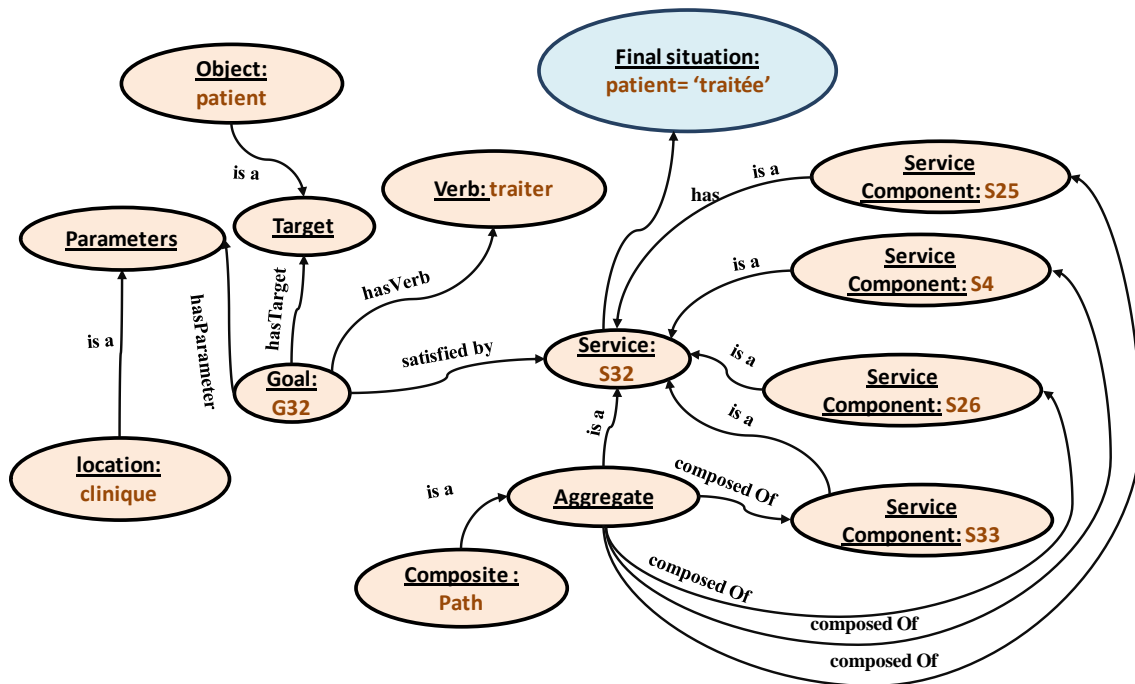


Présentation du service composite S réaliser une visite médicale d'un patient dans iOnto

```
<service name="réaliser une visite médicale d'un patient"
  iServiceConcept="http://.../iServiceOntology#Verb
    http://.../iServiceOntology;#Product
    http://.../iServiceOntology;#Referent
    http://.../iServiceOntology;#final_stat
    http://.../iServiceOntology;#Path
    http://.../iServiceOntology;#ServiceComponent
    http://.../iServiceOntology;#ServiceComponent
    http://.../iServiceOntology;#ServiceComponent
  modelReference="http://.../VerbOntology;#réaliser
    http://.../HealthOntology;#visite médicale
    http://.../HealthOntology;#patient
    http://.../HealthOntology;#visite='réalisée' OR 'reportée'
    http://.../iServiceOntology;#S31
    http://.../ iServiceOntology;#S9
    http://.../ iServiceOntology;#S27
    http://.../ iServiceOntology;#S30">
</service name>
```

Le descripteur du service composite S réaliser une visite médicale d'un patient

1.2.30. Service agrégat S_{32}

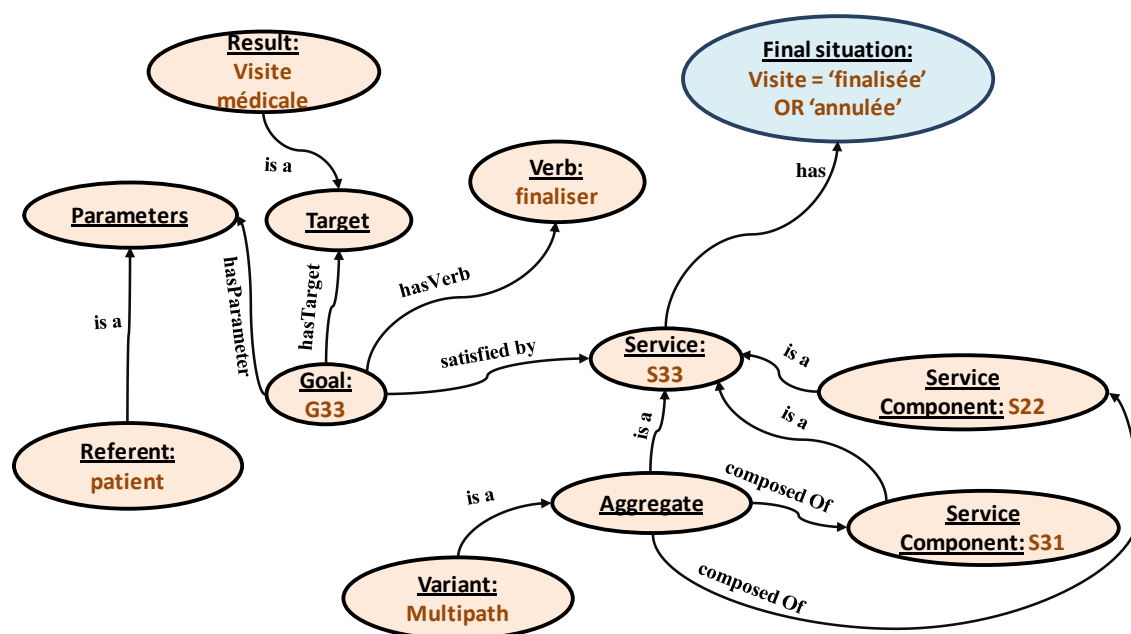


Présentation du service composite $S_{\text{traiter un patient dans la clinique}}$ dans iSOnto

```
<service name="traiter un patient dans la clinique"
  iServiceConcept="http://.../iServiceOntology#Verb
    http://.../iServiceOntology;#Product
    http://.../iServiceOntology;#Referent
    http://.../iServiceOntology;#final_stat
    http://.../iServiceOntology;#Path
    http://.../iServiceOntology;#ServiceComponent
    http://.../iServiceOntology;#ServiceComponent
    http://.../iServiceOntology;#ServiceComponent
    http://.../iServiceOntology;#ServiceComponent
  modelReference="http://.../VerbOntology;#traiter
    http://.../HealthOntology;#patient
    http://.../HealthOntology;#clinique
    http://.../HealthOntology;#patient='traité'
    http://.../iServiceOntology;#S32
    http://.../iServiceOntology;#S25
    http://.../iServiceOntology;#S4
    http://.../iServiceOntology;#S26
    http://.../iServiceOntology;#S33">
</service name>
```

Le descripteur du service composite $S_{\text{traiter un patient dans la clinique}}$

1.2.31. Service agrégat S_{33}



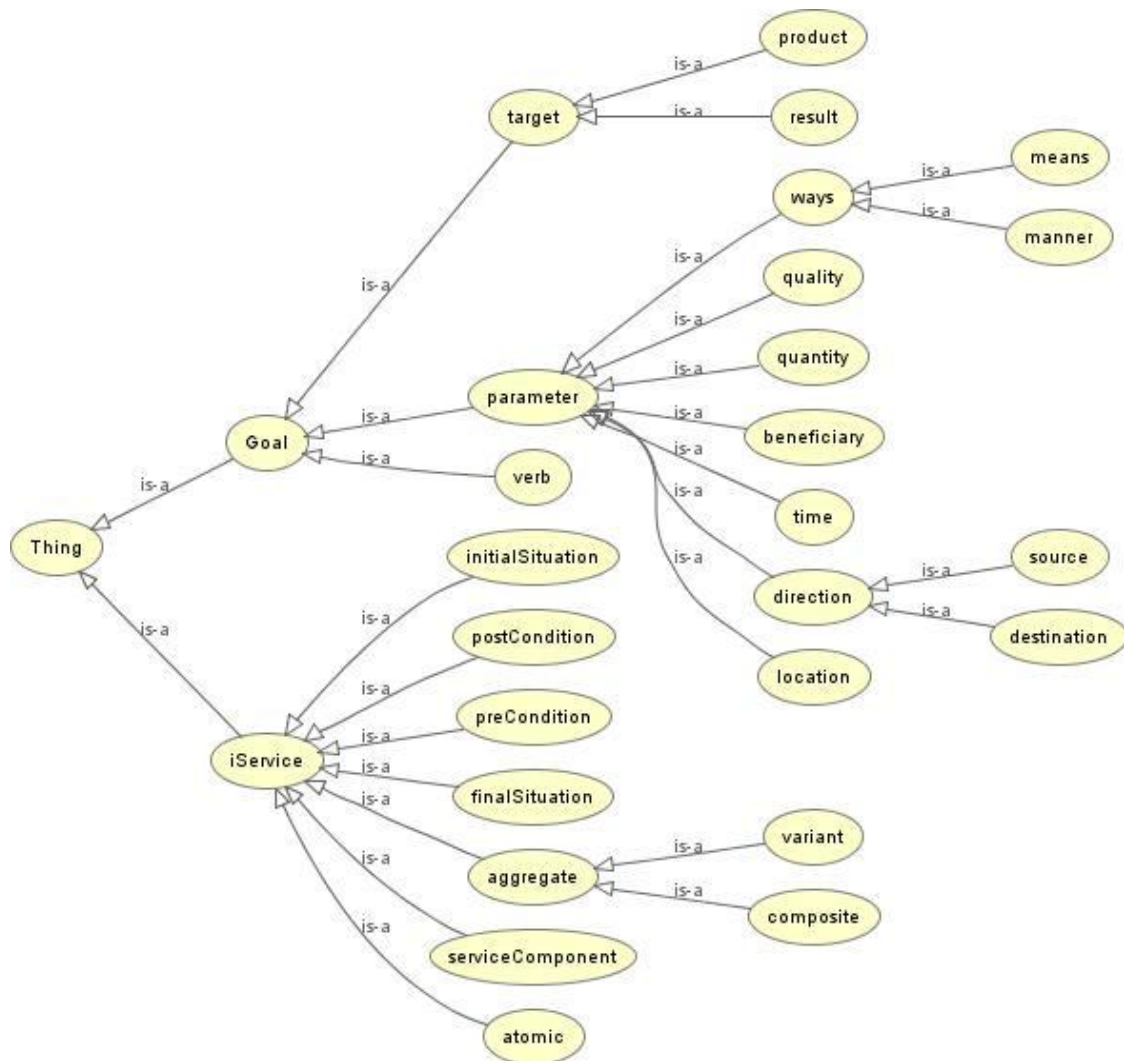
Présentation du service composite S finaliser une visite médicale d'un patient dans iOnto

```
<service name="finaliser une visite médicale d'un patient"
  iServiceConcept="http://.../iServiceOntology#Verb
    http://.../iServiceOntology;#Product
    http://.../iServiceOntology;#Multipath
    http://.../iServiceOntology;#Referent
    http://.../iServiceOntology;#final_stat
    http://.../iServiceOntology;#ServiceComponent
    http://.../iServiceOntology;#ServiceComponent
  modelReference="http://.../VerbOntology;#finaliser
    http://.../HealthOntology;#visite médicale
    http://.../HealthOntology;#patient
    http://.../HealthOntology;#visite = 'finalisée' OR 'annulée'
    http://.../iServiceOntology;#S33
    http://.../ iServiceOntology;#S22
    http://.../ iServiceOntology;#S31">
</service name>
```

Le descripteur du service composite S finaliser une visite médicale d'un patient

ANNEXE B

1. ONTOLOGIE DES SERVICES INTENTIONNELS « *ISONTO* » PAR PROTEGE



2. L'ONTOLOGIE *ISONTO* EN OWL

```

<?xml version="1.0"?>

<!DOCTYPE Ontology [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

```

```

<Ontology xmlns="http://www.w3.org/2002/07/owl#"
xml:base="http://www.semanticweb.org/ontologies/2011/3/Ontology130329168804
6.owl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

ontologyIRI="http://www.semanticweb.org/ontologies/2011/3/Ontology130329168
8046.owl">
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#" />
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#" />
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
  <Declaration>
    <Class IRI="#Goal" />
  </Declaration>
  <Declaration>
    <Class IRI="#aggregate" />
  </Declaration>
  <Declaration>
    <Class IRI="#atomic" />
  </Declaration>
  <Declaration>
    <Class IRI="#beneficiary" />
  </Declaration>
  <Declaration>
    <Class IRI="#composite" />
  </Declaration>
  <Declaration>
    <Class IRI="#destination" />
  </Declaration>
  <Declaration>
    <Class IRI="#direction" />
  </Declaration>
  <Declaration>
    <Class IRI="#finalSituation" />
  </Declaration>
  <Declaration>
    <Class IRI="#iService" />
  </Declaration>
  <Declaration>
    <Class IRI="#initialSituation" />
  </Declaration>
  <Declaration>
    <Class IRI="#location" />
  </Declaration>
  <Declaration>
    <Class IRI="#manner" />
  </Declaration>
  <Declaration>
    <Class IRI="#means" />
  </Declaration>
  <Declaration>
    <Class IRI="#parameter" />
  </Declaration>
  <Declaration>
    <Class IRI="#postCondition" />
  </Declaration>
  <Declaration>

```

```
<Class IRI="#preCondition"/>
</Declaration>
<Declaration>
  <Class IRI="#product"/>
</Declaration>
<Declaration>
  <Class IRI="#quality"/>
</Declaration>
<Declaration>
  <Class IRI="#quantity"/>
</Declaration>
<Declaration>
  <Class IRI="#result"/>
</Declaration>
<Declaration>
  <Class IRI="#serviceComponent"/>
</Declaration>
<Declaration>
  <Class IRI="#source"/>
</Declaration>
<Declaration>
  <Class IRI="#target"/>
</Declaration>
<Declaration>
  <Class IRI="#time"/>
</Declaration>
<Declaration>
  <Class IRI="#variant"/>
</Declaration>
<Declaration>
  <Class IRI="#verb"/>
</Declaration>
<Declaration>
  <Class IRI="#ways"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#composedOf"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasFinalSituation"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasInitialSituation"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasPostCondition"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasPreCondition"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasTarget"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#hasVerb"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#relatedTo"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#satisfiedBy"/>
```

```
</Declaration>
<SubClassOf>
  <Class IRI="#aggregate"/>
  <Class IRI="#iService"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#atomic"/>
  <Class IRI="#iService"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#beneficiary"/>
  <Class IRI="#parameter"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#composite"/>
  <Class IRI="#aggregate"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#destination"/>
  <Class IRI="#direction"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#direction"/>
  <Class IRI="#parameter"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#finalSituation"/>
  <Class IRI="#iService"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#initialSituation"/>
  <Class IRI="#iService"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#location"/>
  <Class IRI="#parameter"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#manner"/>
  <Class IRI="#ways"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#means"/>
  <Class IRI="#ways"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#parameter"/>
  <Class IRI="#Goal"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#postCondition"/>
  <Class IRI="#iService"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#preCondition"/>
  <Class IRI="#iService"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#product"/>
  <Class IRI="#target"/>
</SubClassOf>
```

```

<SubClassOf>
  <Class IRI="#quality"/>
  <Class IRI="#parameter"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#quantity"/>
  <Class IRI="#parameter"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#result"/>
  <Class IRI="#target"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#serviceComponent"/>
  <Class IRI="#iService"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#source"/>
  <Class IRI="#direction"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#target"/>
  <Class IRI="#Goal"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#time"/>
  <Class IRI="#parameter"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#variant"/>
  <Class IRI="#aggregate"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#verb"/>
  <Class IRI="#Goal"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#ways"/>
  <Class IRI="#parameter"/>
</SubClassOf>
<SubObjectPropertyOf>
  <ObjectProperty IRI="#composedOf"/>
  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
  <ObjectProperty IRI="#hasFinalSituation"/>
  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
  <ObjectProperty IRI="#hasInitialSituation"/>
  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
  <ObjectProperty IRI="#hasPostCondition"/>
  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>
  <ObjectProperty IRI="#hasPreCondition"/>
  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
</SubObjectPropertyOf>
<SubObjectPropertyOf>

```

```

    <ObjectProperty IRI="#hasTarget"/>
    <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#hasVerb"/>
    <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#relatedTo"/>
    <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#satisfiedBy"/>
    <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
  </SubObjectPropertyOf>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#composedOf"/>
    <Class IRI="#aggregate"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasFinalSituation"/>
    <Class IRI="#iService"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasInitialSituation"/>
    <Class IRI="#iService"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasPostCondition"/>
    <Class IRI="#iService"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasPreCondition"/>
    <Class IRI="#iService"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasTarget"/>
    <Class IRI="#Goal"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#hasVerb"/>
    <Class IRI="#Goal"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty IRI="#satisfiedBy"/>
    <Class IRI="#Goal"/>
  </ObjectPropertyDomain>
  <ObjectPropertyDomain>
    <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
    <Class IRI="#verb"/>
  </ObjectPropertyDomain>
  <ObjectPropertyRange>
    <ObjectProperty IRI="#composedOf"/>
    <Class IRI="#serviceComponent"/>
  </ObjectPropertyRange>
  <ObjectPropertyRange>
    <ObjectProperty IRI="#hasFinalSituation"/>
    <Class IRI="#finalSituation"/>
  </ObjectPropertyRange>
  <ObjectPropertyRange>
    <ObjectProperty IRI="#hasInitialSituation"/>

```

```
<Class IRI="#initialSituation"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasPostCondition"/>
  <Class IRI="#postCondition"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasPreCondition"/>
  <Class IRI="#preCondition"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasTarget"/>
  <Class IRI="#target"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#hasVerb"/>
  <Class IRI="#verb"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#satisfiedBy"/>
  <Class IRI="#iService"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>
  <Class IRI="#target"/>
</ObjectPropertyRange>
</Ontology>

<!-- Generated by the OWL API (version 3.0.0.1413)
http://owlapi.sourceforge.net -->
```